Project Proposal: H1N1 Vaccine Prediction

Tao Shan, Xiaonan Liu, Yaochen Xiao, Sijie Chen

April 24, 2023

Contents

Summary

Keywords: H1N1 Pandemic, Classification, Machine Learning

This report presents an analysis of an H1N1 vaccination imbalanced dataset using various aspects of the participants' backgrounds information to predict whether participants have received H1N1 flu vaccines. The machine learning classifiers employed include SVM, logistic regression, decision tree, random forest, neural network, and boosting. The study focuses on optimizing the tuning parameters of each model using K-fold cross-validation to enhance the ROC-AUC and PR-AUC scores. The findings reveal that most classifiers achieved an ROC-AUC score of approximately 0.89, with random forest, neural network, and boosting demonstrating a higher PR-AUC of 0.76. After evaluating performance metrics, interpretability, and time complexity, XGBoost, GBM, and random forest are deemed the most appropriate classifiers for this specific dataset.

1 Introduction

The World Health Organization (WHO) declared the outbreak a pandemic on June 11, 2009, naming the virus as influenza A (H1N1)pdm09. In contrast to previous seasonal H1N1 viruses, only a small percentage of young people possessed immunity to the (H1N1)pdm09 virus, according to the US Centers for Disease Control and Prevention (CDC). This new strain, which resulted from a reassortment of genes from human, swine, and avian influenza viruses, quickly spread across the world, infecting millions of people and causing substantial morbidity and mortality. After the (H1N1)pdm09 vaccine was widely available, Schwarzinger (2010) observed a low acceptability of A/H1N1 pandemic vaccination in the French adult population. In light of the current pandemic, we aim to investigate vaccine acceptability and rate interacting with social factors in the United States.

In recent years, machine learning algorithms, particularly classification models, have emerged as powerful tools to identify patterns and gain insights into population behaviour. This report aims to explore the use of classification models, to determine the factors that contribute to an individual's likelihood of receiving the H1N1 vaccine.

Several studies have already investigated the application of machine learning techniques for predicting vaccination uptake. H. Chen et al. (2011) utilized decision trees and logistic regression to identify factors associated with H1N1 vaccination intentions in Hong Kong, while Bish et al. (2011) employed similar methods to explore vaccination behaviour during the H1N1 pandemic in the United Kingdom. More recently, researchers have started to explore the utility of more advanced classification algorithms like XGBoost. For instance, Johnson et al. (2019) applied XGBoost to predict vaccination rates in the United States, demonstrating its efficacy in handling large-scale and complex datasets.

The primary objective of this study is to investigate the relationship between data preprocessing techniques, model architectures, and classification performance. By evaluating a range of preprocessing methods, such as data cleaning, feature selection, feature scaling, and encoding, we aim to identify the optimal preprocessing pipeline for various classification tasks. Additionally, we explored the performance of different model structures.

The remainder of this paper is structured as follows. Section 2 presents the dataset and the data manipulation process. In Section 3, we describe the six methods and models that we employ to analyze the dataset. Section 4 provides the preliminary analysis and results based on our current code. Finally, in Section 5, we discuss the results and outline potential questions that we aim to explore further in the project.

2 Data

The data is provided courtesy of the United States National Center for Health Statistics. The data set is also uploaded to Kaggle and has 6 notebooks (link). The data set contains 39 predictors, 2 outcomes, and around 25 thousand observations focusing on people's opinions about H1N1 and seasonal flu vaccine, social information and whether they had H1N1 and seasonal flu vaccine.

2.1 Data Cleaning

5.99% of data is missing in the entire dataset. To address this issue, we decided to exclude columns with over 40% missingness. Next, we used the MICE (Multivariate Imputation By Chained Equations) (Mera-Gaona et al. 2021) method to handle the numerical missing data. The algorithm imputes missing values with plausible estimates based on observed data using a series of regression models. For each variable with missing values, a regression model is created with other variables as predictors, and then used to impute missing values. The process is repeated for a fixed number of iterations or until convergence, generating multiple complete datasets which are analyzed using the same statistical model. By imputing missing values multiple times using different models and initial guesses, MICE captures the uncertainty in the imputation process and produces more accurate estimates and standard errors for the parameters of interest. For categorical columns, we employed mode imputation, grouping by social factors including sex, race, and age group. This approach was taken as a part of our report to ensure that the missing data was handled effectively and that our analysis was based on a complete dataset.

2.2 Data Exploration

Upon analyzing the data, we did not observe any strong correlations between individuals receiving the vaccine and other selected variables. (Figure 1 (left)) The majority of observations, approximately 80%, indicate that individuals did not receive the vaccine by early 2010, revealing the data is inbalanced. (Figure 1 (right)) However, we did find that individuals who were more concerned about H1N1, older in age, and believed in the vaccine's effectiveness had a higher likelihood of being vaccinated. Furthermore, individuals who received a doctor's recommendation were also more likely to get vaccinated. This finding is expected since individuals with a doctor's recommendation may have pre-existing medical conditions or be more health-conscious.Surprisingly, we did not find a clear relationship between vaccination rates and geographical regions. Overall, the data suggests that individual characteristics, such as age, beliefs, and recommendations, have a more significant impact on vaccine uptake than geographic location.

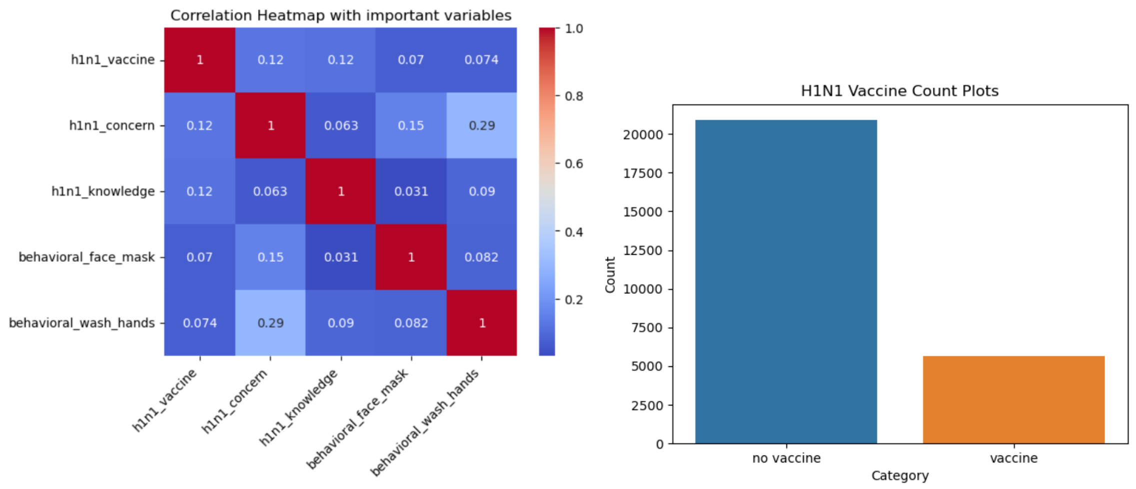

Figure 1: Correlation Heatmap (left). H1N1 Vaccine Count Plots (right)

2.3 Feature Generation

Feature generation is a technique employed to enhance model performance by utilizing existing features to generate new, informative data (Li et al. 2020). By generating new features, we can potentially uncover hidden patterns and improve the overall predictive capability of machine learning models. We created several new features based on logical calculations and domain knowledge, which we believe will be useful. First, we generated a variable representing the interaction between age group and chronic medical conditions, as we assume that older individuals are more likely to have chronic diseases. We also created another interaction variable for healthcare workers and health insurance. Additionally, we generated a list representing the difference in opinion ratings between the H1N1 pandemic and seasonal flu, as well as an indicator denoting whether a household has at least one child.

2.4 Data Preprocessing

Given that most predictors in the dataset are categorical, we converted these variables into a numerical format using one-hot encoding and ordinal encoding to enable effective interpretation and processing by machine learning algorithms. Many of these variables represent opinions or levels of knowledge, which inherently have a ranking. In such cases, each unique category is assigned an integer value based on its rank or position in the order, facilitating model assessment. Additionally, we transformed the other categorical variables into binary indicators excluding the least frequent label in each variable. In addition to encoding, we applied min-max scaling to standardize the range of features. Overall, these data preprocessing steps, including encoding and scaling, ensure that our dataset is in an appropriate format for analysis by various machine learning algorithms, enhancing their ability to accurately model the underlying patterns in the data.

2.5 Feature Selection

Feature selection methods play a crucial role in reducing the dimensionality of training data, which can lead to faster and more accurate predictors (Guyon and Elisseeff 2003). In our study, we employed the Chi-square test to identify the most relevant features. The Chi-square test is an appropriate measure for determining the dependence between categorical variables based on their frequency distribution in a contingency table. As our predictors are mostly categorical, the Chi-square test is a good fit. By ranking the features according to their relevance, we selected those with Chi-square test scores greater than 0.1, ensuring that our model focuses on the most influential features while minimizing the risk of overfitting and improving computational efficiency.

3 Methods and Models

3.1 Evaluation Metrics and Tuning

Our project is an imbalanced classification problem, the majority class is typically referred to as the negative outcome (i.e. respondents didn't receive H1N1 flu vaccine), and the minority class is referred to as the positive outcome (i.e. respondents received H1N1 flu vaccine). We will use ROC-AUC and Precision-Recall AUC (PR-AUC) as our performance metrics.

ROC curve is a plot that summarizes the performance of a binary classification model on positive class. The x-axis indicates the False Positive Rate and the y-axis indicates the True Positive Rate. It considers both sensitivity=true positive rate and specificity=1-false positive rate across various decision thresholds, which is useful for comparing different models.

Precision-Recall AUC (Rana 2019) is particularly useful when the minority class is of higher interest, which is a positive outcome for H1N1-vaccine. It focuses on the balance between precision and recall. This metric is less sensitive to the majority class and can provide a more accurate representation of model performance on imbalanced datasets.

We tune parameters by evaluating potential values and identifying candidate values visualized in the results section with line plots. We employed RandomizedSearchCV in sklearn first, a technique that randomly selects a subset of hyperparameter combinations from a predefined search space. Then, we were able to efficiently explore a wide range of parameter values, thereby reducing the computational time and resources required compared to an exhaustive search. Next, we tuned the single common parameters each time, to check the optimal range. After identifying a region in the hyperparameter space. Then, we applied GridSearchCV to evaluate combinations of parameters within a narrower space around the optimal values.

3.2 Support Vector Machines

Support Vector Machines (SVM) is a powerful supervised learning algorithm that tries to find the optimal hyperplane that separates the two classes with the maximum margin. We plan to tune Kernel (linear, poly, rbf) and C: Regularization parameters. In each Kernel, we use cross-validation with GridSearchCV and RandomizedSearchCV to find the optimal value of C.

3.3 Logistic Regression

Logistic regression is a widely-used statistical method for modelling the relationship between a dichotomous target variable and one or more predictor variables. The model applies a logit transformation to the odds of the target event to predict the probability of a positive outcome. In our dataset, logistic regression is a highly suitable classifier given its binary nature. The model is known for its simplicity and interpretability, as it generates coefficients for each independent variable, allowing for a better understanding of their impact on the outcome.

One of the key strengths of Logistic Regression is its simplicity and interpretability. It provides coefficients for each independent variable, which can be used to understand the impact of each feature on the outcome. We will fit and compare the performance of logistic regression models using various regularization techniques and strengths, along with hyperparameter tuning (solver, max-iterations).

3.4 Tree Method

Decision Tree is a non-parametric, hierarchical model that recursively splits the data into subsets based on the values of input features, resulting in a tree-like structure. We will fit and compare Decision Tree with cost complexity pruning and Bagging.

We first train an initial decision tree on the training data without pruning, it seems that there're too many nodes and likely overfitting the training data. Then we apply cost complexity pruning. To find optimal alpha, we use node impurity and K-fold cross-validation via scikit-learn to conduct combined CV losses with a function of alpha. The minimum alpha is the optimal alpha of cost complexity pruning. We refit a pruned tree that is clearer.

Main hyperparameters:

- max_depth: the maximum depth of the tree
- min_samples_split: the minimum number of samples required to split an internal node
- min_samples_leaf: the minimum number of samples required for a leaf node
- max_features: the number of features to consider for the best split

After a random search of a combination of parameters that gives us a range of parameter values, we check each parameter with cv error and roc plots, then apply grid search for different criteria to figure out the adjusted tree using the reduced parameter range.

For Bagging, we adjust the model by considering the complexity (min_samples_leaf, max_features, etc.) of the tree, and loss function with node impurity (Gini index or entropy).

3.5 Random Forest

Random Forest is an ensemble learning technique that builds on the idea of decision trees. Instead of relying on a single decision tree, it creates a collection of decision trees, each constructed using a random subset of the available features and a random subset of the training data. By combining the predictions of all the decision trees, the Random Forest is more robust to overfitting and other types of errors that can occur with a single decision tree. This ensemble approach reduces the impact of individual tree errors and enhances the overall generalization performance.

The main package used for the model is RandomForestClassifier from scikit-learn. The model is first trained in the default setting. Next, we'll tune parameters including the maximum depth of the tree and the minimum number of samples required to split an internal node and to be at a leaf node (min-samples-split and min-samples-leaf) to narrow down the range of tuning. In addition, the class-weight parameter is particularly relevant for imbalanced datasets, as it assigns greater weight to the minority class, an aspect worth exploring in this case. After that, we conduct a grid search with cross-validation to find the optimal combination that results in the highest accuracy while maintaining the AUC score.

3.6 Neural Network

Neural networks are a class of machine learning models that are inspired by the structure and function of the human brain. They consist of layers of interconnected nodes, or neurons, that perform mathematical operations on the input data to generate output values. As of today, Neural networks have demonstrated impressive performance in various domains and have been proven to be very powerful in terms of classification tasks.

Training a neural network involves feeding it with a set of labeled examples and adjusting its weights and biases to minimize the error between its predicted output and the actual output. This process is typically performed using an optimization algorithm such as stochastic gradient descent.

We will be implementing the multi-layer perceptron (MLP) network using the MLPClassifier package in scikit-learn as our baseline classifier. The default settings of MLPClassifier will be used as the starting point for our analysis. To improve the performance of the classifier, we will first tune the hyper-parameters individually, starting with the number of layers (hidden_layer_sizes), the strength of the L2 regularization term (alpha), and the batch size. We will experiment with different values for each hyper-parameter and evaluate the impact on the classifier's cross-validation error and AUC score. After having the possible optimal range for each parameter, we will then perform a grid search with cross-validation to identify the optimal hyperparameter settings.

3.7 Gradient Boosting

Gradient boosting, a powerful ensemble learning technique, involves the sequential construction of decision trees. In contrast to the random forest, each tree of GBM focuses on the errors made by the previous one and uses shallow decision trees as weak learners that can reduce overfitting. GBM is a strong predictive model that excels in handling diverse data types and complex relationships (Friedman 2001).

After first training with the default setting, we tune parameters including not only max_depth, min_samples_split, min_samples_leaf, max_features, but also the learning rate that determines contribution of each tree to the ensemble, and subsample (subsampling rate), which represents the fraction of samples used for constructing each tree. Then we apply grid search with cross-validation to identify the optimal hyperparameter settings based on reduced optimal range.

3.8 XGBoost

XGBoost is a highly scalable and efficient machine learning system for tree boosting, created as an open-source package by Tianqi Chen and Carlos Guestrin in 2016. It has gained widespread popularity and proven success, as evidenced by its use in 17 of the 29 winning solutions published on Kaggle's blog during 2015. In comparison, the second most popular method, deep neural networks, was used in only 11 solutions. (T. Chen and Guestrin 2016)

XGBoost's core strength lies in its ability to minimize a regularized objective function, which combines a convex loss function with a penalty term for model complexity. This optimization approach enables the algorithm to iteratively train new trees that predict the residuals or errors of prior trees. These new trees are combined with previous ones to create the final prediction. (T. Chen and Guestrin 2016) It also works well when data is nonlinear, non-monotonic, or with segregated clusters.

Specifically for our dataset, XGBoost has great performance, robustness, and flexibility making it a strong choice for classifying people's vaccination status. Our dataset has lots of categorical type of data, and XGBoost is a tree-based ensemble method which well suited for categorical type data. Also, XGBoost incorporates L1 and L2 regularization to reduce the over-fitting problem. XGBoost which also has a great model explain-ability by feature importance score.

We fine tune the model using the following main and common parameters:

- learning_rate: This determines the step size shrinkage used in the update to prevent overfitting. Typical values range from 0.01 to 0.3.
- max_depth: The maximum depth of a tree, which controls the complexity of the model. Typical values range from 3 to 10.
- min_child_weight: The minimum sum of instance weight needed in a child to prevent overfitting.
- gamma: A regularization parameter that controls the minimum loss reduction required to make a split.
- subsample: The fraction of samples to be used for each boosting round, which helps in preventing

overfitting

We aim to fine-tune these parameters to achieve optimal results. By doing so, we can enhance model performance and ensure that our dataset's model is comparable to other models.

4 Results and Analysis

4.1 SVM

The Support Vector Machine (SVM) model is a robust and versatile algorithm widely used for classification and regression tasks. In our analysis, we fine-tune the model using different kernel types, as well as the C and gamma parameters. To address the issue of imbalanced data, we utilize the class-weight parameter set to "balanced". After extensive experimentation, we select the following optimal configuration for our SVM model: Kernel to be Radial basis function (RBF), C = 0.631, and Gamma = 0.126.

To facilitate the parameter tuning process, we employ grid-search methods in conjunction with graphical representations. Graphs help us identify the most suitable parameter ranges for our grid-search exploration. By visualizing the relationship between parameter values and model performance, we can effectively narrow down our search space and efficiently pinpoint the optimal configuration.

With our fine-tuned SVM model, we achieve the performance metrics with ROC-AUC = 0.893, and PR-AUC = 0.737. These results demonstrate the effectiveness of our chosen SVM configuration, yielding a strong performance in terms of both Receiver Operating Characteristic (ROC) curve and Precision-Recall (PR) curve. The high ROC-AUC and PR-AUC scores indicate that our SVM model is capable of accurately classifying data points, even in the presence of class imbalance.

4.2 Logistic Regression

The logistic regression model we worked with initially had an accuracy of 86.2% and an AUC score of 89.2%. To enhance its performance and minimize potential bias, we tuned several hyperparameters, including solver, penalty, regularization strength (C), and maximum iteration. We began by investigating the relationship between the model's CV error and ROC-AUC score with the regularization strength parameter C. Figure 2 revealed that as we increased C, the CV error decreased, and the ROC-AUC score increased until C exceeded 1. We observed that the CV error was the lowest when C was between 0 and 0.5. Using a grid search with 50 values between exp(-4) to exp(1.3), we narrowed the range of C and other hyperparameters to fine-tune the regularization penalty and improve the model's performance.

The final hyperparameters we selected were saga for the solver algorithm, lasso for the penalty, a maximum iteration of 10,000, and a regularization strength of 0.837. With these hyperparameters, the model achieved a ROC-AUC score of 89.2% and a PR-AUC score of 74.6%. Our careful selection and tuning of hyperparameters enabled us to improve the logistic regression model's performance (ROC-AUC) by 3%.

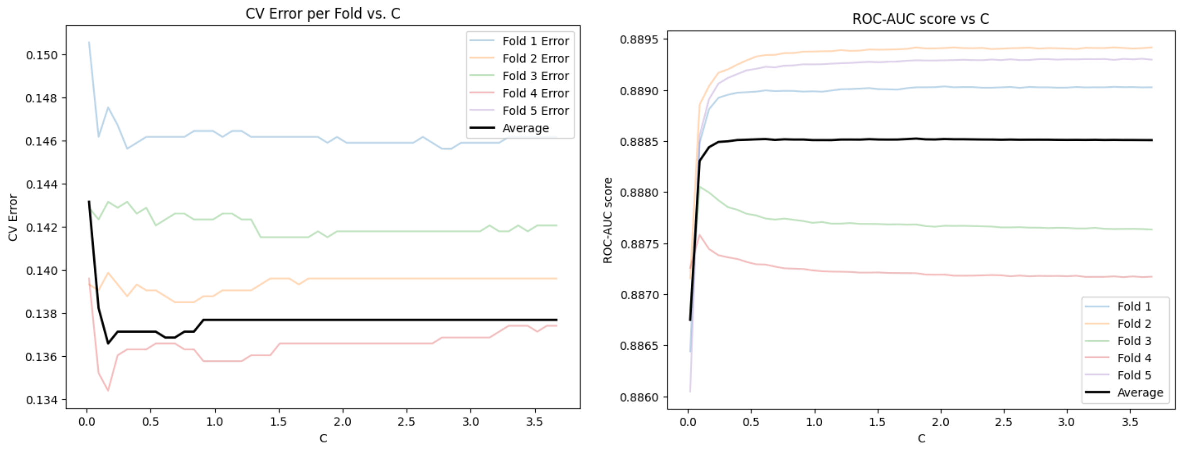

Figure 2: Logistic Regression CV error vs C (left). Logistic Regression ROC-AUC vs C (right)

4.3 Decision Tree

We examine an initially complex tree, with our baseline decision tree (default parameters) achieving ROC-AUC 0.706 and PR-AUC 0.601. We then proceed to conduct cost complexity pruning to identify alphas with CV losses. Upon determining the optimal alpha value of 0.0004, we refit and plot a more comprehensible pruned tree.

To enhance the performance, we first apply random search with 5-fold cross-validation of main hyperparameters in 3.5 for gini and entropy criterion. We find out that gini with roc 0.878 prauc 0.731, and entropy with roc 0.879 prauc 0.736, which are not significant differences, then we will continue the following analysis using both criteria.

We set a relatively narrow range for each main parameter based on random search results. According to the approximate optimal value of cv loss and roc plots for each parameter, we further reduce the range of combined parameters in the grid search. We attain a roc 0.876 PR-AUC 0.708 with entropy, and ROC-AUC of 0.880, along with a PR-AUC of 0.730 with gini. Using Bagging that combined with the tuned tree, we attain 0.888 ROC-AUC and 0.753 PR-AUC.

4.4 Random Forest

The default random forest classifier yields an accuracy of 85.7% and an AUC score of 0.887. To further evaluate the model's robustness, a 5-fold cross-validation is performed, with the AUC score as the metric of interest. The results show minimal variation between each fold, indicating the model's stability and consistency in its performance. This is an encouraging finding that suggests the model's reliability in making predictions on new data.

In the individual tuning phase, most of the parameters do not have a significant impact on the model performance except for the maximum depth, which should be at least 10 to avoid underfitting. Upon conducting a thorough grid search, the results suggest limiting the maximum depth to 17, changing min_samples_leaf/split to 2 and 6, and setting the class-weight option as None. The model's accuracy rises to 86.1% and the AUC score increases to 0.895. (Figure 3 (left)) Furthermore, it is worth mentioning that the grid search for the random forest appears to be very time expensive. The search checked more than 1000 combinations and took a good 6 hours to complete.

Regarding feature importance, Figure 3 (right) indicates that the three most crucial factors are related to doctor's recommendations, respondents' seasonal vaccination status, and respondents' opinions on vaccination, which align with commonly perceived factors.

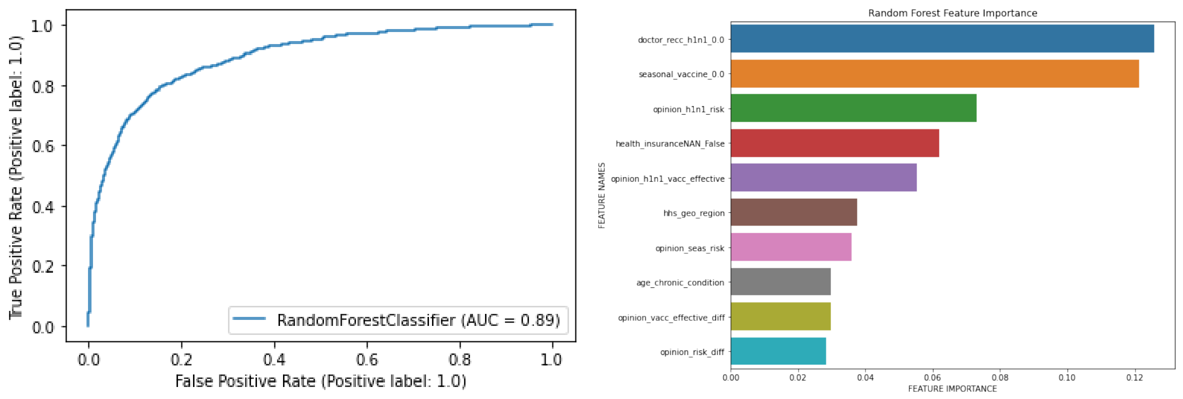

Figure 3: Random Forest Tuned Roc (left). Random Forest Feature Importance (right)

4.5 Neural Network

To ensure convergence, the maximum number of epochs is set to be 400, and the baseline MLP classifier with 100 layers gives an accuracy of 85.4% and an AUC score of 0.864.

During the tuning phase, Figure 4 (left) suggests that large layer sizes lead to overfitting and thus the appropriate number should be around 10. On the other hand, alpha and batch size seem to have little impact on either of the evaluation metrics.

After the grid search, the program determines that the optimal settings are alpha: 0.008, a batch size of 256, and 6 hidden layers (8 layers in total). And the optimized classifier achieves an accuracy of 86.2% and an AUC score of 0.892. Figure 4 (right) shows the confusion matrix of the results.

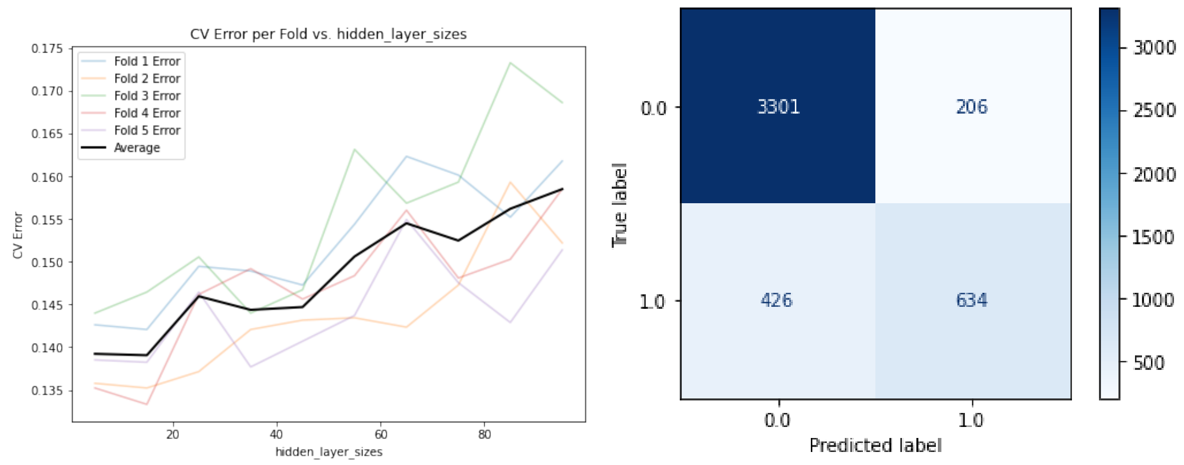

Figure 4: Neural Network CV Error per Fold (left). Neural Network Confusion Matrix (right)

4.6 Gradient Boosting

We examine the baseline performance of gradient boosting using n_estimators set to 100 and random_state at 42, which yields a 0.897 ROC-AUC and 0.768 PR-AUC. To improve the performance, we initially focus on a common range for key parameters such as learning rate (0.01, 0.3), max_features (1, X_train.shape[1] + 1), and subsample (0.5, 1), thereby generate an optimal combination of parameters using randomsearchCV. As a result, we achieve a PR-AUC of 0.775 with subsample=0.684, learning_rate=0.059, max_features=12, and other parameters, representing a roughly 1% improvement.

During the tuning phase, we analyze the ROC-AUC score and CV loss plots for each common hyperparameter. For instance, max_depth exhibits the lowest CV loss and highest ROC score within the range of (3,6), as depicted in Figure 5 (left) across all 5-fold cross-validation. This range is then considered as a narrowed-down candidate for the grid search's parameter combination. By achieving a balanced trade-off between model complexity and generalization, we obtain max_depth = 3, subsample = 0.65, learning_rate= 0.17, etc., resulting in a 0.898 ROC-AUC and 0.767 PR-AUC with increased recall. Note that GBM constructs trees sequentially, grid search becomes time-consuming, then we use less combination range with less total fits, so we may find that grid is not as good as random search. Consequently, we opt for a smaller combination range with fewer total fits, which may cause the grid search to be less effective.

According to the feature importance in Figure 5 (right), the three most important factors are H1N1 flu vaccine was recommended by the doctor, respondents' seasonal vaccination status, and respondents' opinion about risk of getting sick with H1N1 flu without vaccine.

4.7 XGBoost

In this study, we aimed to optimize the performance of our XGBoost model by fine-tuning its hyperparameters. Our baseline XGBoost model, without any hyperparameter optimization, achieved a ROC-AUC score of 0.860. To enhance the model's performance, we focused on optimizing three key hyperparameters: max depth, learning rate, and the number of estimators. By systematically adjusting each of these parameters, we were able to improve the ROC-AUC score to 0.898, which represents a significant increase in predictive accuracy.

Figure 6 shows the relationship between the parameter and ROC-AUC score by 5-fold cross-validation.

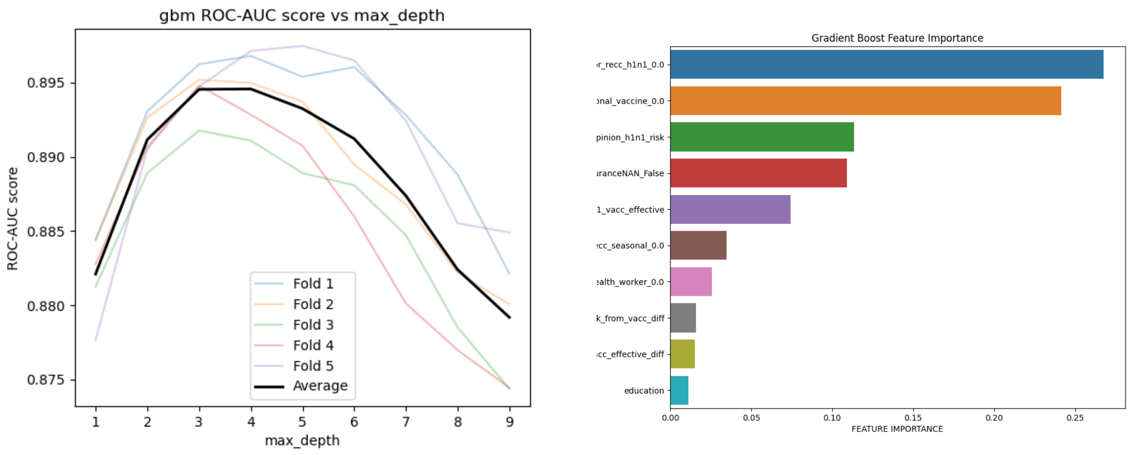

Figure 5: GBM ROC-AUC vs Max depth (left). GBM Feature Importance (right)

Furthermore, the graphs display the performance of each fold, allowing us to examine the model's consistency across different data splits.

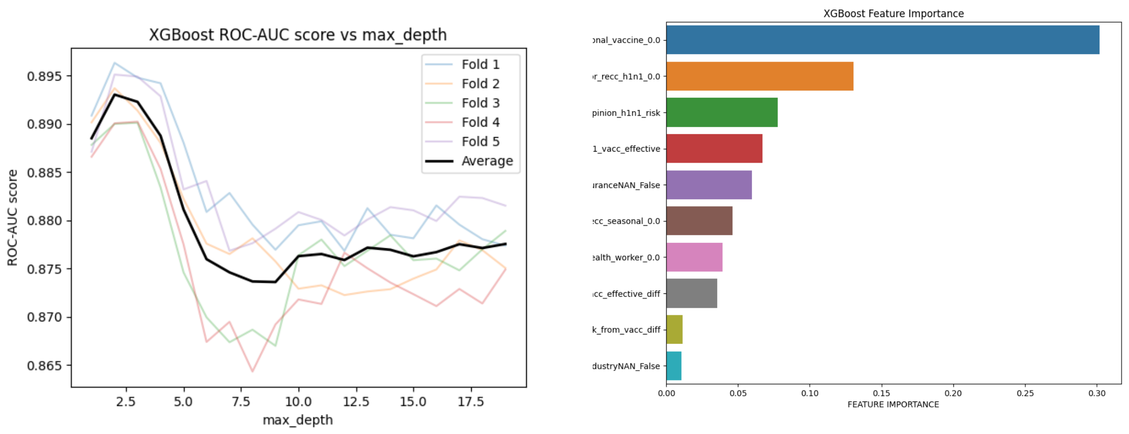

Figure 6: XGBoost ROC-AUC score vs Max Depth (left). XGBoost Feature Importance (right)

Table 1: Performance comparison of different models

Model	ROC-AUC	PR-AUC
SVM	0.893	0.737
Logistic Regression	0.892	0.745
Decision Tree	0.88	0.73
Bagging	0.888	0.753
Random Forest	0.894	0.768
Neural Network	0.893	0.763
GBM	0.897	0.767
XGBoost	0.898	0.768

5 Discussion

In this section, we discuss the results of the performance comparison of different models as presented in Table 1. The models have been evaluated based on two important metrics: ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) and PR-AUC (Precision-Recall Area Under the Curve), which are commonly used for evaluating classification performance. The table reveals that gradient boosting models (XGBoost and GBM) and the RandomForest model provide the best performance for this classification task, while Decision Tree has the lowest performance. Almost all models are in a similar and acceptable range of performance. For ROC-AUC, the difference between highest and lowest are only 0.02.

It is crucial to consider not only performance metrics but also factors such as interpretability, training time, and computational resources when selecting a model. Although machine learning techniques like Neural Networks can be effective for classification tasks, they may not always be the best choice depending on the specific problem and dataset. Similarly, while both SVM and Logistic Regression have decent performance, they might struggle with more complex patterns in the data. In conclusion, the results indicate that XGBoost, GBM, and RandomForest are the most suitable models for this particular classification problem, but other factors must be taken into account before making a final decision.

At present, we have utilized the Chi-square test as the criterion for feature selection. However, feature selection encompasses a wide array of techniques that can be employed to identify the most relevant features for our model. Forward and Backward selection and regularization methods with Lasso or Ridge are all potential as our further work. In our project, we could explore these alternative feature selection techniques to identify the best subset of features for our model, potentially improving its performance and interpretability.

Other than compare the model's performance, based on a general understanding, the variable health insurance could be a significant predictor of vaccination rates. However, with 46% of its data missing, it may lead to poor prediction accuracy. In future research, we may explore alternative methods for handling missing data and further investigate this topic in the project.

For the model performance, we see the confusion matrix has a large false negative compared with true positive, which makes the performance of Recall=TP/(TP+FN) score worse. In order to get a comprehensive understanding of model performance for the imbalanced dataset, we use a combination of metrics.

Note that the features of our dataset are more categorical rather than numeric. We find that our models don't have a significant improvement after model tuning. It seems like the training data is noisy or has outliers that affect the model's performance. Besides, grid search may not have explored parameter combinations that address the class imbalance effectively. In future research, we may consider cost-sensitive learning, undersampling the majority class, oversampling the minority class, or the Synthetic Minority Over-sampling Technique (SMOTE) (Bunkhumpornpat, Sinapiromsaran, and Lursinsap 2012) to address the class imbalance and improve model performance.

6 Conclusion

This report describes a study aimed at investigating the factors that contribute to an individual's likelihood of receiving the H1N1 vaccine during a pandemic. We begin by providing background information on the H1N1 pandemic and its impact on global health, and introduce the dataset and its features. To analyze the dataset, we employed seven different machine learning algorithms, which are described in Section 3 of the report. We conducted a rigorous tuning process to optimize each model and evaluated their performance using ROC-AUC and PR-AUC scores. In our analysis, we observed that some models showed greater improvements in performance than others, and we discussed potential reasons for these differences.

In conclusion, we suggest XGBoost, GBM or RandomForest algorithms analyze this dataset considering interpretability and running time. As for future work, we plan to explore alternative methods for handling missing data, additional feature selection techniques such as Forward and Backward selection, and a combination of metrics and consider techniques to improve model performance.

Overall, Our study aims to contribute to machine learning algorithms' application in public health research. We hope that the results of our study will provide valuable insights into vaccine acceptability and rate, which will help guide policy decisions and public health initiatives aimed at increasing vaccination uptake.

Reference

Bish, Alison, Lucy Yardley, Angus Nicoll, and Susan Michie. 2011. "Factors Associated with Uptake of Vaccination Against Pandemic Influenza: A Systematic Review." *Vaccine* 29 (38): 6472–84.

Bunkhumpornpat, Chidchanok, Krung Sinapiromsaran, and Chidchanok Lursinsap. 2012. "DBSMOTE: Density-Based Synthetic Minority over-Sampling TEchnique." *Applied Intelligence* 36 (4): 664–84. https://doi.org/10.1007/s10489-011-0287-y.

Chen, Hong, Daniel Zeng, Ping Yan, and Fang Chen. 2011. "Vaccination Intention and Influential Factors in Case of an H1n1 Pandemic: A Survey of the Hong Kong General Public." *Health* 3 (6): 378–82.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785.

Friedman, J. H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics* 29 (5). https://doi.org/10.1214/aos/1013203451.

Guyon, Isabelle, and André Elisseeff. 2003. "An Introduction to Variable and Feature Selection." *J. Mach. Learn. Res.* 3 (null): 1157–82.

Johnson, Lydia, Kieran O'Neill, Brittany Landon, and Jovana Stojanovic. 2019. "Predicting Vaccination Uptake Using the XGBoost Algorithm." In *Proceedings of the Annual International Conference on Machine Learning and Data Science*, 123–30.

Li, Li, Haolin Yang, Yueming Jiao, and Kuo-Yi Lin. 2020. "Feature Generation Based on Knowledge Graph." *IFAC-PapersOnLine* 53 (5): 774–79. https://doi.org/10.1016/j.ifacol.2021.04.172.

Mera-Gaona, Miguel, Uwe Neumann, Richard Vargas-Canas, and Dorian M López. 2021. "Correction: Evaluating the Impact of Multivariate Imputation by MICE in Feature Selection." *PloS One* 16 (12). https://doi.org/10.1371/journal.pone.0261739.

Rana, Shahzad Ali Khan; Zeeshan Ali. 2019. "Evaluating Performance of Software Defect Prediction Models Using Area Under Precision-Recall Curve (AUC-PR)." In *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*, pp. 249–267.

Schwarzinger, Rémi AND Cortarenoda, Michaël AND Flicoteaux. 2010. "Low Acceptability of a/H1n1 Pandemic Vaccination in French Adult Population: Did Public Health Policy Fuel Public Dissonance?" *PLOS ONE* 5 (4): 1–9. https://doi.org/10.1371/journal.pone.0010199.