Final Report

“Prediction of COVID-19 Vaccine Side effects”

Team 08

Department of Statistics	2015-15347	Wonhyeong Choe
Department of Mathematical Sciences	2016-11797	Kihyun Han

1. Introduction

1.1. Research Background

Since the rate of the population who completed COVID-19 vaccine injection exceeded 78% in South Korea, South Korea has become one of the countries having the highest vaccination rate. Accordingly, the public health authority has alleviated the social distancing restriction and implemented the “With-Corona” policy, thus encouraging many people to participate in social activities, which reduced due to COVID.

However, there are controversies concerning the vaccine’s side effects. Although there are a lot of cases of diseases reported through the press, the authority does not acknowledge the causal relation and correlation of these instances and vaccination. In fact, according to Korea Disease Control and Prevention Agency, a number of side effect instances such as thrombosis (혈전증), anaphylaxis (아나필락시스), myocarditis (심근염), and pericarditis (심낭염) occurred. Also, the deceased instances have recorded 894, but there are only 6-7 cases where its correlation is admitted.

We study the classification by analyzing the factors that have an influence on deaths or life-threatening illnesses with respect to factors such as age, sex, allergies, and COVID vaccine manufacturers – Pfizer, Moderna, and Janssen. We expect that the prediction model makes it possible for people to understand and be aware of the potential danger of vaccination in a statistical and quantitative fashion.

1.2. Data Description

We obtained data from Kaggle, which was collected from VAERS (Vaccine Adverse Event Report System). We analyzed the cases of COVID vaccine side effects reported from January to November 2021.

2. Methods

2.1. Data Pre-processing

First, we selected variables that can affect the occurrence of death and life-threatening illnesses from the data. Then, we removed observations in which some variables were not recorded and duplicated data. This made the data reduce to 432,816 instances.

In the data, symptoms were recorded as a type of MedDRA (Medical Dictionary for Regulatory Activities) term. Generally, the MedDRA dictionary contains a five-level hierarchy (Figure 1), and the symptoms in the data are written as "Preferred Term" (PT), which accounts for the fourth hierarchy. We assigned the corresponding "System Organ Class" (SOC), the highest hierarchy with 27 elements, for each of the symptoms. Then we transformed into 27 dummy variables representing whether each individual suffered the symptom in a particular SOC.

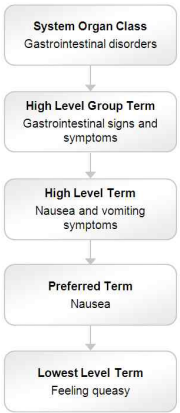

Figure 1 MedDRA symptom hierarchy example[3]

SYMPTOM1	SYMPTOM2	SYMPTOM3	SYMPTOM4	SYMPTOM5
Dysphagia	Epiglottitis			
Anxiety	Dyspnoea			
Chest discomfort	Dysphagia	Pain in extremity	Visual impairment	
Dizziness	Fatigue	Mobility decreased		
Injection site erythema	Injection site pruritus	Injection site swe	Injection site warmth	
Pharyngeal swelling				
Abdominal pain	Chills	Sleep disorder		
Diarrhoea	Nasal congestion			
Vaccination site erythen	Vaccination site pruri	Vaccination site swelling		
Rash	Urticaria			
Blood pressure decreas	Chest pain	Chills	Confusional stat	Decreased appetit
Dyspnoea	Fatigue	Feeling abnorma	Head discomfort	Headache
Heart rate decreased	Heart rate increased	Hypertension	Injection site pai	Musculoskeletal c
Nausea	Pain	Pain in extremity	Paraesthesia oral	Pyrexia
SARS-CoV-2 antibody t	SARS-CoV-2 test negative			
Ear pain	Hypoaesthesia			
Abdominal pain upper	Dizziness	Dysgeusia		
Blood pressure increase	Chest discomfort	Heart rate increased		

SYMPTOM1	SYMPTOM2	SYMPTOM3	SYMPTOM4	SYMPTOM5	SYMPTOM6	SYMPTOM7	SYMPTOM8
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1

Figure 2 Classification for symptom variable. "Preferred term" (Left) was transformed into dummy variables indicating which "System Organ Class" the symptom is included (Right).

Since 'History' and 'Allergies' variables were entered as sentences by individuals, which is unstructured data, we applied natural language processing to detect keywords of each variable. To be specific, for the 'History' variable, we selected the terms and prefixes that appeared more than 500 times in the data. Then we classified the keywords with respect to their medical interpretation into 23 classes. For instance, 4 keywords "kidney", "nephritis", "nephr-", and "CKD" are related to kidney diseases, so we put these as the same class. Then we develop 23 dummy variables representing whether an individual had history with associated classes. In the similar manner, we make 'Allergy' dummy variables to represent whether each included individual has the specific allergy.

CUR_ILL	HISTORY	ALLERGIES
None	None	Pcn and bee venom
Patient residing at nursing facility. See patients cha	Patient residing at nursing facility. See patients chart.	"Dairy"
None	None	Shellfish
Na	NA	Na
NA	NA	iodine (shellfish) has epipen
None	None	None
none	Hashimoto's thyroiditis, Hypertension, depression	Sulfa antibiotics, azithromycin, adhesive in band-aids or tape
NA	NA	jackfruit
Covid-19 (symptom onset 12-16-20 negative test :	Obesity Anxiety	Dust mites Zoloft Wellbutrin Buspar
none	Graves Disease	penicillin, toradol, methimazole
None	None	None
none	None	None
None	Migraines, PMDD	Sulfa, steri strips, adhesive bandages
NA	Depression	None
NA	NA	NA
NA	NA	Sulfa; Wellbutrin
None	Covid 19 infection with lingering symptoms to include chronic dy	None
None	None.	Ceftiaxone (Rocephin)
None	Asthma, ADHD	Seasonal/Environmental Allergies
Bacterial Upper Respiratory infection(2 weeks ago)	Anxiety	Morphine

HISTORY21	HISTORY22	HISTORY23	ALLERGY1	ALLERGY2	ALLERGY3
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	1	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 3 Keyword Analysis for 'History' and 'Allergy' variables. Sentence form (Left) was transformed into dummy variables indicating the inclusion of each keyword (Right).

In addition, we created two dummy variables representing the vaccine manufacturer, and 11 dummy variables representing each month of vaccination. Also, we wanted to find whether the temperature of the state where a person is vaccinated would affect the rate of illnesses, so we added "MEAN_TEMP" variable, which indicates the mean temperature of a state by month.

In order to consider the non-linear effect of continuous variables, say "MEAN_TEMP, "AGE", and "NUMDAYS", we consider the quartiles of each variable as knots of polynomial models of degree

three. If this model is applied, we obtain the cubic spline model for these continuous variables. The process yields the 18 explanatory parameters since cubic spline models generate six variables for each variable, excluding the intercept term. Note that the quartiles of a continuous variable “HOSPDAYS” are all zero; thus, we did not apply the above transformation.

For the response variable, we made a dummy variable, “THREAT”, to represent whether a person has a death or life-threatening illnesses. The explanatory and response variables are summarized as follows.

- Explanatory variables: 113 (Continuous: 19, Discrete: 94)

Variables	Description	Variable Type
SYMPTOM1 - SYMPTOM27	Adverse symptoms in MedDRA Term	Transformed into 27 dummy variables representing highest hierarchy.
MODERNA - PFIZER	Vaccine manufacturer	Transformed into 2 dummy variables representing Pfizer, Moderna, and Janssen(default)
VAX_DOSE_SERIES	Number of doses administered	Discrete variable representing the first and the second dose
MEAN_TEMP_1 - MEAN_TEMP_3_4	Mean temperature of the state by month	Transformed into 6 continuous variables with cubic spline terms of three knots
AGE_1 - AGE_3_4	Age in years	Transformed into 6 continuous variables with cubic spline terms of three knots
SEX	Sex	Dummy variable
HOSPITAL	Hospitalized	Dummy variable
HOSPDAYS	Number of days hospitalized	Continuous variable
DISABLE	Disability	Dummy variable
JAN – NOV	Month of vaccination	Transformed into 11 dummy variables representing each month (default: December)
NUMDAYS_1 – NUMDAYS_3_4	Number of days from vaccination date to onset date	Transformed into 6 continuous variables with cubic spline terms of three knots
HISTORY1 - HISTORY23	Chronic or long-standing health conditions	Transformed into 23 dummy variables representing history
BIRTH_DEFECT	Congenital anomaly or birth defect	Dummy variable
ALLERGY1 - ALLERGY26	Allergies to medications, food, or other products	Transformed into 26 dummy variables representing history

- Response variable: 1 (Discrete: 1)

Variables	Description	Variable Type
THREAT	Death or life-threatening illnesses	Dummy variable

2.2. Boosting Algorithms

We discuss the boosting algorithms that are used in our research in greater detail, specifically gradient boosting, AdaBoost, and XGBoost. These methods are well-known to show high performance in practice.

2.2.1. Gradient Boosting Algorithm

We aim to minimize the loss function $L(y, f(x))$, by sequentially growing the trees. Gradient boosting algorithm employs M base trees with n data and investigates the pseudo-residuals at each m-th step. It computes the function that minimizes the residual sum of squares and multiply with the learning rate λ_m. The pseudocode of the algorithm is as follows. In practice, computing the multiplier (2-3) is often removed, and λ_m is pre-assigned for the computational advantage.

1. Initialize constant $\widehat{f^{(0)}}(x) = \widehat{f_0}(x) = \widehat{\gamma_0} = arg\ \min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
2. For $m = 1,2, \cdots, M,$ do:
   2-1. Compute the pseudo-residuals: $r_{i,m} = \left[-\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\widehat{f^{(m-1)}}(x)}$
   2-2. Find the optimal m-th tree: $\widehat{\phi_m} = \arg\min_{\phi} \sum_{i=1}^{n} [r_{i,m} - \phi(x_i)]^2$
   2-3. Compute the multiplier: $\lambda_m = arg\ \min_{\lambda} \sum_{i=1}^{n} L\left(y_i, \widehat{f_{m-1}}(x_i) + \lambda \widehat{\phi_m}(x_i)\right)$
   2-4. Update the model with: $\widehat{f_m}(x_i) = \widehat{f_{m-1}}(x_i) + \lambda_m \widehat{\phi_m}(x_i)$
3. Output $\widehat{f_M}(x)$

2.2.2. AdaBoost Algorithm

AdaBoost is inherently based on gradient boosting algorithm, but for each step m, AdaBoost minimizes the exponential loss $\mathrm{E}(f) = \sum_{i=1}^{n} e^{-y_i f(x_i)}$ where the response variable $y_i \in \{-1,1\}$. It finds the optimal y_i It is known that the algorithm assigns a higher weight to the data that are misclassified by previous classifiers, thus improving the performance.

1. Initialize weight $w_i^{(1)} = 1$
2. For $m = 1,2, \cdots, M$, do:

2-1. Compute weight $w_i^{(m)} = e^{-y_i \widehat{f_{m-1}(x_i)}}$ for $m \neq 1$

2-2. Find the optimal tree classifier $k_m(x) \in \{-1,1\}$ that minimizes the misclassified weighted sum $\sum_{y_i \neq k_m(x_i)} w_i^{(m)}$

2-3. Compute the multiplier: $\gamma_m = \arg\min_{\gamma} E\left(\widehat{f_{m-1}} + \gamma \cdot k_m\right) = \arg\min_{\gamma} E\left(\widehat{f_{m-1}} + \gamma \cdot k_m\right)$

2-4. Update the model with: $\widehat{f_m}(x_i) = \widehat{f_{m-1}}(x_i) + \gamma_m k_m(x_i)$

3. Output $\widehat{f_M}(x)$

2.2.3. XGBoost Algorithm

XGBoost algorithm applies the second-order Taylor approximation of the loss function. Here, we may add the regularization term $\sum_{m=1}^{M} \Omega(\phi_m) = \gamma T + \frac{1}{2}\lambda|w|^2$ to the loss function in order to achieve a higher performance, where T is the size of the tree and w indicates the score in the corresponding leaves. Also, the learning rate α appears in the XGBoost algorithm.

1. Initialize constant $\widehat{f^{(0)}}(x) = \widehat{f_0}(x) = \widehat{\gamma_0} = arg\ \min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
2. For $m = 1,2, \cdots, M,$ do:

2-1. Compute the gradients $\widehat{g_m}(x_i)$ and hessians $\widehat{h_m}(x_i)$:

$$\widehat{g_m}(x_i) = \left[\frac{\partial L\left(y_i, f(x_i)\right)}{\partial f(x_i)}\right]_{f(x)=\widehat{f^{(m-1)}}(x)}$$

$$\widehat{h_m}(x_i) = \left[\frac{\partial^2 L\left(y_i, f(x_i)\right)}{\partial f(x_i)^2}\right]_{f(x)=\widehat{f^{(m-1)}}(x)}$$

2-2. Find the optimal m-th tree $\widehat{\phi_m}$:

$$\widehat{\phi_m} = \arg\min_{\phi} \sum_{i=1}^{n} \frac{1}{2} \widehat{h_m(x_i)} \left[-\frac{\widehat{g_m}(x_i)}{\widehat{h_m}(x_i)} - \phi(x_i)\right]^2 + \Omega(\phi)$$

2-3. Update the model with: $\widehat{f_m}(x_i) = \widehat{f_{m-1}}(x_i) + \alpha\widehat{\phi_m}(x_i)$

3. Output $\widehat{f_M}(x)$

2.3. Model Design

We applied random undersampling in order to deal with the imbalance discovered in the response variable. The cases of death or life-threatening illnesses were 12,230, which was relatively smaller than the rest (about 400,000). Therefore, we sampled 12,230 observations that did not result in death or life-threatening illnesses in order to make balanced data. Then we analyzed with 24,460 observations.

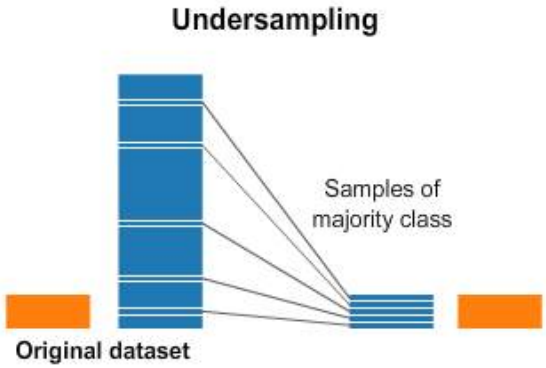

Figure 4 Illustration for undersampling[5]

We considered two models, a model with all predictor variables (main-model) and a reduced model excluding posterior predictor variables (sub-model). We denote posterior predictor variables as information that appears after the vaccination was operated, such as side effect symptoms. In other words, in the sub-model, we analyze the probability of death and life-threatening illnesses before the vaccine injection. Specifically, in the sub-model, we removed "SYMPTOMS1-27", "HOSPITAL", "HOSPDAYS", "NUMDAYS_1 - 3_4". Therefore, the sub-model contains $113 - 27 - 1 - 1 - 6 = 78$ predictor variables.

The methods we applied are logistic regression (LR), linear discriminant analysis (LDA), K-nearest neighbors (KNN), naïve Bayes classifier, random forest, gradient boosting, AdaBoost, and XGBoost. For all methods, we implemented 10-fold cross-validation (CV) and found the best subset of predictor variables that minimizes the CV error. We regard the CV error as a measure that compares the performance of the methods. We add that the accuracy is identical to the balanced accuracy (BA) since the numbers of the case and the control are equal.

3. Results

3.1. Logistic Regression (LR)

We developed a classifier by setting the threshold value as 0.5. In the main-model with LR method, the best model is attained when p=54 with CV error rate 0.144. By comparing the fitted values and the observed values of the test set, we observe that the proportion classified as 1 (death or life-threatening illnesses) highly increases as the estimated probability increases, especially near 1. This implies that the model fit is well operated.

We split the data into the training set and the test set. After training, we draw the ROC curve and get

the AUC value, 0.922. Here, we marked the point on the ROC curve with the closest to the top left corner. The point was (0.860, 0.855), with the distance 0.445 from the corner.

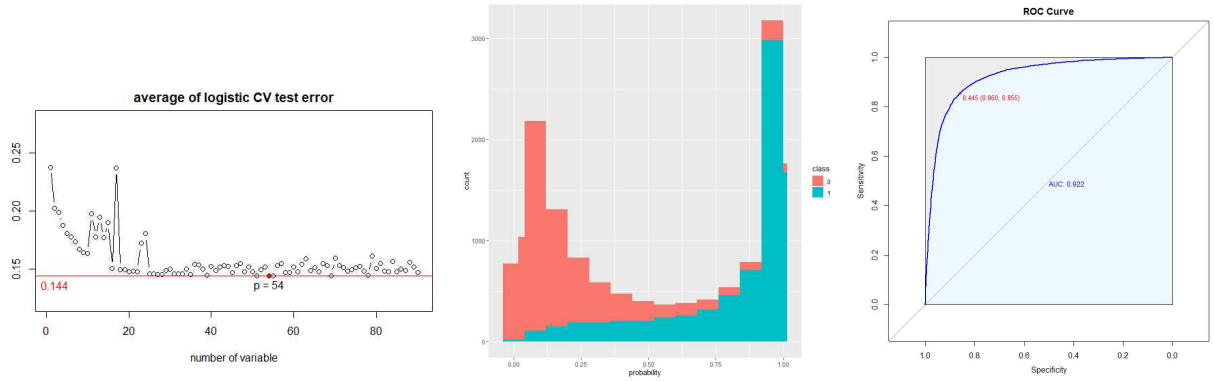

Figure 5 Logistic regression fitting of the main-model. Cross-validation errors for validation selection (Top), observed value vs. fitted value (Bottom left), ROC curve (Bottom right).

In the sub-model, the best model is attained when p=55 with CV error rate 0.288. Compared to the main-model, the sub-model's fitted probability values mostly lie in the middle of 0 and 1. This implies that the model is not accurate. In the ROC curve, we get the AUC value of 0.771. The point at the closest to the top left corner is (0.759, 0.669), with the distance 0.495 from the corner. We observe that the AUC and the distance are larger in the sub-model, implying lower accuracy. We may interpret the result as follows: the sub-model does not include the symptoms, which may have high correlation with the response variable.

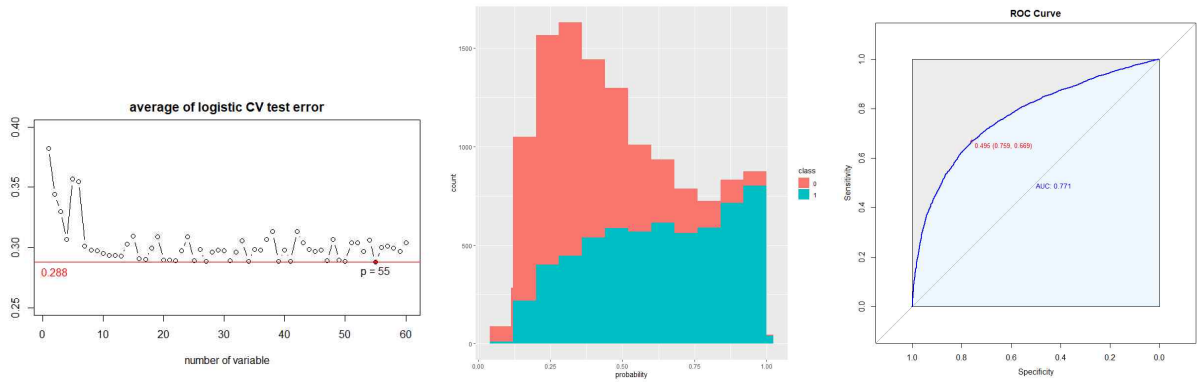

Figure 6 Logistic regression fitting of the sub-model. CV errors for validation selection (Top), observed value vs. fitted value (Bottom left), ROC curve (Bottom right).

3.2. Linear Discriminant Analysis (LDA)

For LDA, we set the threshold value to 0.5 as in the LR case. The following plot shows the CV error

of linear discriminant analysis. As the number of predictor variables increases, the CV error of LR and LDA models exhibit a decreasing trend. The number of variables of the lowest CV error is 88 and 58 for the main-model and the sub-model, respectively. Since the CV errors are higher than those of the logistic regression, we conclude that these methods will have weaker performance than the logistic regression model.

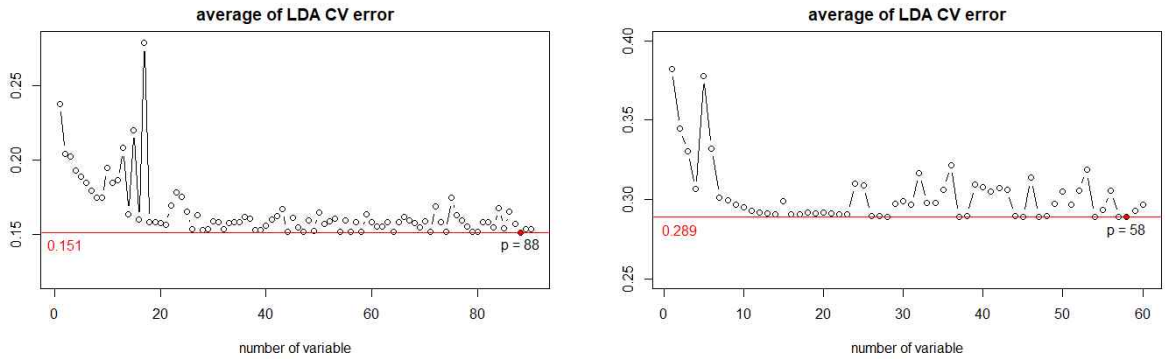

Figure 7 Linear discriminant analysis CV errors for the main-model (Left) and the sub-model (Right)

3.3. K-Nearest Neighborhood (KNN)

We applied the K-nearest neighborhood (KNN) method for different values of hyperparameter K = 1, 3, 5, 7, 9, 11, and 13. In contrast to the previous models, KNN models have the optimal number around 20. The minimal CV errors were 0.15 when K = 11, 13.

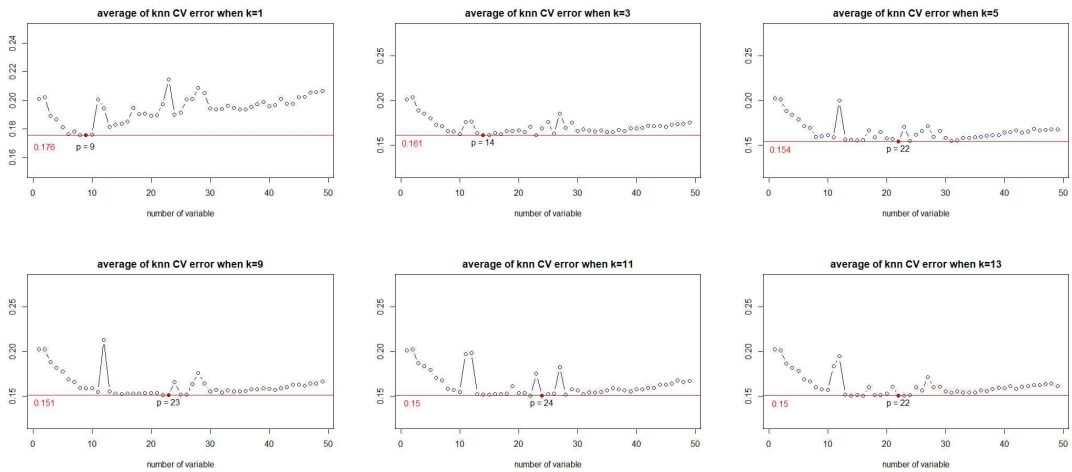

Figure 8 KNN CV errors in the main-model

In the sub-model, the KNN models have the lowest CV errors when K = 13, and the error is 0.298. The errors in both main-model and sub-model are higher than the above analysis methods, so the KNN

method is not the best prediction model.

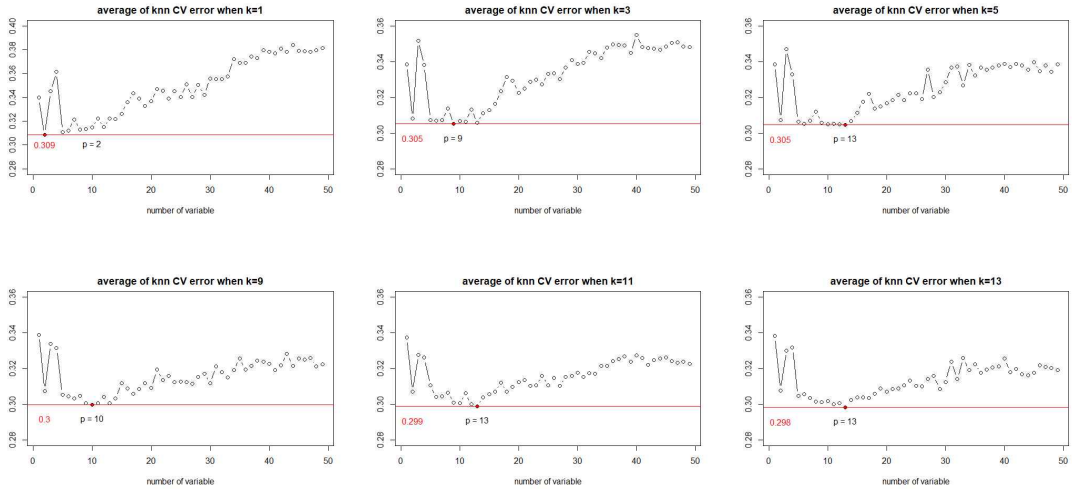

Figure 9 KNN CV errors in the sub-model

3.4. Naïve Bayes classifier

The naïve Bayes classifier's CV errors are 0.167 and 0.339 in the main-model and the sub-model, respectively. The numbers of parameters are 19 and 12. The naïve Bayes method show poor performance since the model has a strong assumption that the observations are independent.

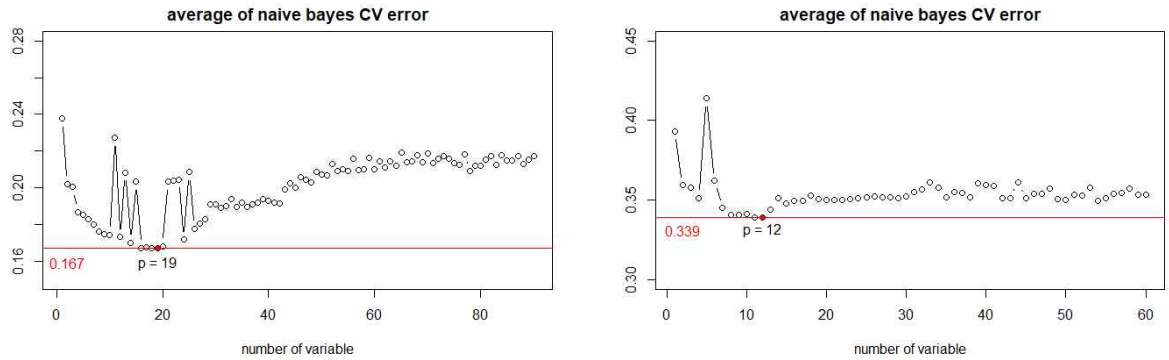

Figure 10 naïve Bayes classifier CV errors in the main-model (Left) and the sub-model (Right)

3.5. Random Forest

The random forest method's CV errors are 0.133 and 0.291 in the main-model and the sub-model, respectively. The numbers of parameters are 32 and 4 for both models. The random forest method shows the lowest CV error for the main-model, compared to the models so far.

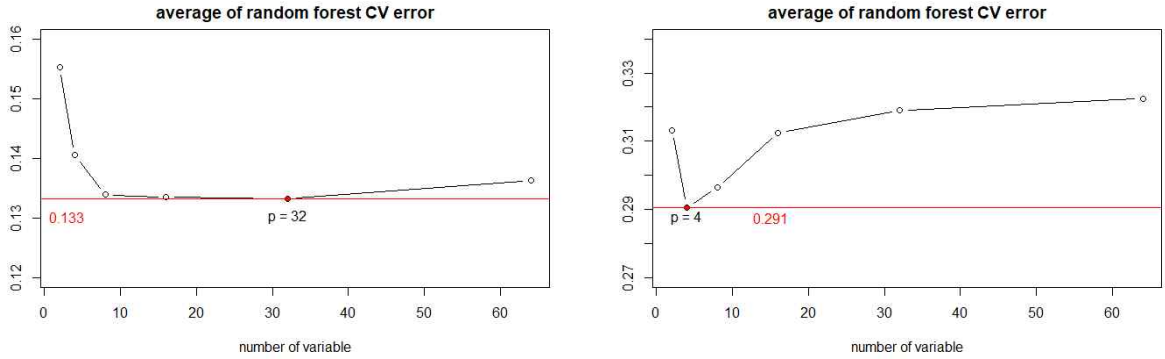

Figure 11 Random forest CV errors in the main-model (Left) and the sub-model (Right)

3.6. Gradient Boosting

In the gradient boosting method, we fix the value of the learning rate λ. The CV error for the gradient boosting method is minimized at iteration 2000 and $\lambda = 0.05$ in the main-model. The CV error is 0.1447.

Table 1 Gradient boosting CV errors for the main-model

#Iter. \ λ	0.1	0.05	0.025	0.0125
500	0.1452576	0.1470973	0.1561325	0.169215
1000	0.1450531	0.1455846	0.1473017	0.1562551
2000	0.1451349	0.1447670	0.1450531	0.1465658
4000	0.1459935	0.1451349	0.1448078	0.1449714

For the sub-model, the CV errors for the gradient boosting method is minimized at iteration 4000 and the learning rate $\lambda = 0.025$. The CV error is 0.2885.

Table 2 Gradient boosting CV errors for the sub-model

#Iter. \ λ	0.1	0.05	0.025	0.0125
500	0.2895748	0.2910057	0.2942764	0.3002044
1000	0.2890025	0.2899019	0.2904742	0.2937040
2000	0.2888798	0.2893704	0.2894113	0.2906787
4000	0.2885119	0.2894930	0.2885119	0.2894113

3.7. AdaBoost

In the AdaBoost method, we use different maximum depths of the tree, say 5, 10, 20, and 30. This contributes to the different complexity of the model. In the main-model, the CV error is minimized at iterations 45 and the maximum depth 20. The CV error value is 0.1363.

Table 3 AdaBoost CV errors for the main-model

#Iter. \ Depth	5	10	20	30
5	0.1607522	0.1548242	0.1574407	0.1539657
15	0.1463614	0.1411693	0.1417416	0.1430090
45	0.1424366	0.1419460	0.1363859	0.1402289
100	0.1382257	0.1388389	0.1370809	0.1374489

3.8. XGBoost

In the XGBoost method, we ignore the L_1 shrinkage and use different L_2 shrinkage parameter λ=0.005, 0.01, 0.03, 0.09, 0.27. Then the minimal CV value for the main-model was 0.128 when the iteration was 2700 and λ=0.01. Moreover, for the sub-model, we get the minimal CV error, 0.287, when the iteration was 2700 and λ=0.005. We discover that the values for both models are the lowest among the methods that we analyze in the research.

Table 4 XGBoost CV errors for the main-model

#Iter. \ λ	0.005	0.01	0.03	0.09	0.27
100	0.1549469	0.1469747	0.1348324	0.1295585	0.1291496
300	0.1422322	0.1356092	0.1290270	0.1287817	0.1327473
900	0.1323794	0.1291496	0.1289452	0.1302535	0.1387572
2700	0.1284546	0.1282911	0.1318070	0.1360998	0.1389616

Table 5 XGBoost CV errors for the sub-model

#Iter. \ λ	0.005	0.01	0.03	0.09	0.27
100	0.2982829	0.2972199	0.2910874	0.2876124	0.2932951
300	0.2952984	0.2916598	0.2880212	0.2903515	0.3070318
900	0.2875715	0.2869582	0.2908013	0.3008585	0.3215453
2700	0.2872444	0.2896565	0.3012264	0.3186835	0.3353229

4. Conclusions

4.1. Best Classifier

We developed classifiers that predict the probability of serious side effects of COVID-19 vaccines using various methods. We concluded that the prediction performance was the highest with the XGBoost method for both the main-model and the sub-model.

Table 6 Summary of CV errors for the main-model and the sub-model

Method	LR	LDA	KNN	Naive Bayes	Random forest	Gradient boosting	AdaBoost	XGBoost
Main-model	0.144	0.151	0.15	0.167	0.133	0.144	0.136	0.128
Sub-model	0.288	0.289	0.298	0.339	0.291	0.288	-	0.287

However, the prediction accuracy values were significantly different between the two models, as can be seen from the following ROC curves. In particular, the AUC values of the main-model and the sub-model are 0.934 and 0.776, while the closest distances of the ROC curves from the top left corner are 0.434 and 0.498, respectively. We expect that this phenomenon comes from the correlation between the life-threatening diseases and the posterior variables, which is naturally existent in the data. For instance, if someone suffered from the side effects and went to the hospital, then this person's probability of dying or having severe illnesses must be high. Therefore, the factors that are not considered in the sub-model may result in higher accuracy.

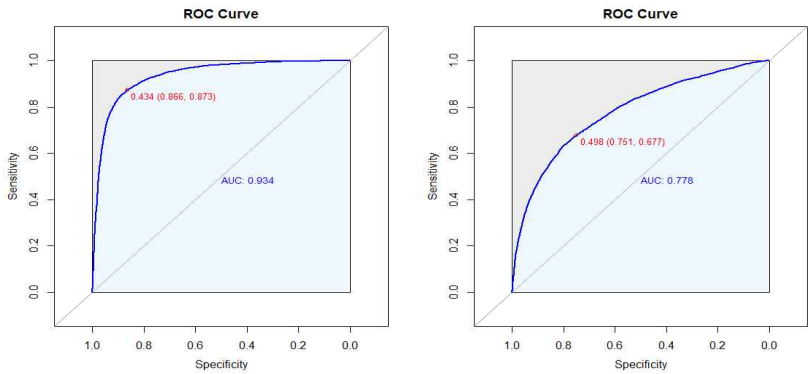

Figure 12 ROC curves of the main-model (Left) and the sub-model

4.2. Predicted Values for the life-threatening diseases

With the sub-model, we can predict the probability of getting the death or life-threatening illnesses

from the health condition of individuals and the vaccine manufacturers. The sub-model indicates the probability without information on symptoms or hospitals, so the predicted value can be practical for people before vaccination.

Since we applied undersampling for the data to get the 12,230 samples of "THREAT" = 0, we may overestimate the predicted probability if we use the value obtained from the obtained XGBoost model. In other words, if a predicted probability for a person is 0.5, then, due to the existence of other 400,000 instances, we cannot directly deduce that this person will die or suffer a severe illness with a probability of 0.5 due to vaccination. Thus, we trained the full data on an XGBoost model in keeping with the hyperparameters obtained from the above analysis.

In the following table, "Healthy" stands for women younger than 30 and do not have any disability, medical history or allergies. "Standard" stands for women in their forties and not disabled but have at least two medical history, or allergies. "Not Healthy" stands for women aged between 66 and 75 with disability and had four or more medical histories or allergies. Then we discover that the Pfizer vaccine has the highest probability for the "Healthy" people, whereas the Janssen vaccine has the highest for "Standard" and "Not Healthy" groups. "Moderna" vaccine can be regarded as the safest among these vaccines since it has the smallest probability. Moreover, we observe that the difference of the probabilities by the health conditions of individuals is significantly large. Therefore, the initial health condition can be a major factor that affect the severeness of the side effects of vaccination.

We need to stress that the values stated in the table are not the actual probability of getting death or life-threatening illnesses after getting vaccinated. The data is not based on all people who received COVID-19 vaccine injection but on the side effect report of 432,816 individuals. In the same time period, say Janu, 236,167,566 people were vaccinated in the United States, which is about 545 times larger than the observations. Hence, we need to multiply 1/545 = 0.0018, on average, to each value in the table to attain the probability of the life-threatening illnesses irrelevant to the occurrence of side effects. If we collect more information on the number of all types of vaccinated people that are considered in the table, then it would be possible to get a far more accurate prediction of the probability.

Table 7 Probability of getting death or life-threatening illnesses after having symptoms

	Pfizer	**Moderna**	**Janssen**
Healthy	0.010700	0.009241	0.010674
Standard	0.017274	0.015310	0.021769
Not Healthy	0.289828	0.273562	0.309278

[References]

[1] COVID-19 Vaccine Adverse Reactions (VAERS) Dataset, https://www.kaggle.com/landfallmotto/covid19-vaccine-adverse-reactions-vaers-dataset?select=vaers_jan_sep_2021.csv

[2] VAERS Data Use Guide, https://vaers.hhs.gov/docs/VAERSDataUseGuide_November2020.pdf

[3] MedDRA Hierarchy, https://www.meddra.org/how-to-use/basics/hierarchy

[4] 박근우 and 정인경. "이분형 자료의 분류문제에서 불균형을 다루기 위한 표본재추출 방법 비교" 응용통계연구 32, no.3 (2019) : 349-374.

[5] Satyam Kumar, “5 Techniques to work with Imbalanced Data in Machine Learning”, https://towardsdatascience.com/5-techniques-to-work-with-imbalanced-data-in-machine-learning-80836d45d30c

[6] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine.." Ann. Statist. 29 (5) 1189 - 1232, October 2001. https://doi.org/10.1214/aos/1013203451

[7] Rojas, Raúl (2009). "AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting". Freie University, Berlin.

[8] Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM. pp. 785–794.

[9] T Chen, T He, M Benesty, “Package ‘xgboost’”, https://cran.r-project.org/web/packages/xgboost/xgboost.pdf

[10] Alfaro, Esteban; Gamez, Matias and Garcia, Noelia, “Package ‘adabag’”, https://cran.r-project.org/web/packages/adabag/adabag.pdf