A Random Forest Prediction Model for Forecasting H1N1 Influenza Vaccine and Season Influenza Vaccine Uptake Using Individual and Collective Behaviors

04/05/2023

1 Methodology

1.1 Research design

This study used a quantitative research design to develop a random forest prediction model for predicting H1N1 influenza vaccination and seasonal influenza vaccination using individual and collective behaviour. The main objective was to create an accurate and reliable predictive model to help public health officials and policy makers understand the factors that influence vaccination rates and develop targeted interventions to improve vaccine coverage and to better prepare vaccines to meet vaccination needs in areas with different circumstances.

Unlike many traditional approaches to this type of problem, such as the use of principal component analysis, this experiment uses machine learning methods to solve the problem. Reasons for choosing a machine learning approach could be explained from the following perspectives:

1. **Complexity of vaccination decisions**:
   Vaccination decision making is a multifaceted process involving a variety of individual, social and collective factors. Traditional statistical methods, such as regression analysis, may not capture the complex interactions and non-linear relationships between these factors. Machine learning algorithms, such as random forests, are able to model complex relationships and interactions in high-dimensional data, making them more suitable for this research.

2. **Dealing with high-dimensional data**:
   The dataset used in this study contains a large number of features, ranging from individual characteristic statistics to community area indicators. Traditional statistical methods may struggle to handle high-dimensional data, leading to problems such as over-fitting or multicollinearity. Machine learning algorithms, such as random forests, can effectively handle high-dimensional data and reduce the risk of overfitting by employing techniques such as bagging and feature randomisation.

3. **Robustness to noise**:
   Real-world data, such as the data used in this study, is often noisy, with missing values, measurement errors or outliers. Machine learning algorithms, such as random forests, are known for their robustness to noise and their ability to handle missing data and outliers without significantly degrading prediction accuracy.

4. **Model interpretability**:
   Although some machine learning models may be considered 'black boxes', random forest models are relatively more interpretable than deep learning. This is critical for public health researchers and policy makers, as understanding the underlying relationships between variables can inform targeted intervention strategies.

5. **Flexibility and adaptability**:
   Machine learning models such as random forests can be easily updated and

retrained as new data becomes available, allowing continuous improvement and adaptation to changing trends in vaccination and public behaviour. This adaptability is critical for effective public health interventions in a rapidly changing environment.

In summary, the decision to use machine learning methods in this study, specifically the Random Forest algorithm, was based on its ability to handle complex, high-dimensional data, its robustness to noise and its ability to model complex interactions between features. In addition, the model's interpretability, feature importance analysis and adaptability make it ideal for understanding the factors that influence H1N1 and seasonal influenza vaccination and for informing targeted public health interventions

1.2 Random forest

Random Forest (RF) is a versatile and widely used ensemble learning method in the field of machine learning and data mining. It was introduced by Leo Breiman in 2001, building on the principles of bagging and decision trees. Random Forest is an ensemble of decision trees, which means it combines multiple tree models to improve the overall prediction accuracy and reduce overfitting. It works for both classification and regression tasks and can handle large datasets with high dimensionality effectively. Decision trees, bagging and feature randomization are the three most important parts when building a random forest.

Decision Trees: A decision tree consists of nodes, branches, and leaves. The nodes represent input features, the branches represent decision rules, and the leaves represent the final output, i.e., class labels for classification tasks or continuous values for regression tasks. Decision trees are constructed by recursively splitting the input space based on the best feature split, which maximizes the information gain (or other specified criteria). However, individual decision trees are prone to overfitting, especially when grown to their maximum depth.

Bagging : Bagging or called Bootstrap Aggregating is an ensemble technique that aims to improve the stability and accuracy of a base model, typically a decision tree, by creating multiple versions of the model using different subsets of the training data. These subsets are obtained by sampling the original dataset with replacement (i.e., bootstrapping). Each bootstrapped dataset is used to train a separate base model, and the predictions of all models are combined through voting (for classification) or averaging (for regression) to obtain the final output. Bagging reduces the overfitting problem of individual decision trees by averaging out the model variance.

Feature Randomization: In addition to bagging, Random Forest introduces feature randomization to increase the diversity of individual decision trees. At each node split during the tree construction, a random subset of features is considered for the split rather than evaluating all available features. This process prevents the trees from being too similar and reduces the correlation between them, further improving the overall model performance.

The pseudo-code of the Random Forest algorithm is shown below:

Algorithm 1: Random Forest Training

Input: Training data X_{train}, y_{train}, number of trees $n\_estimators$, maximum features $max\_features$, maximum depth max_depth

Output: Random Forest model $Forest$

initialize empty forest $Forest$; **for** $i = 1$ *to* $n\_estimators$ **do**

  $q_{bootstrap}, y_{bootstrap} \leftarrow$ bootstrap sample of X_{train} and y_{train};

  $Tree \leftarrow$ build decision tree using $X_{bootstrap}$, $y_{bootstrap}$, $max\_features$, and max_depth; add $Tree$ to $Forest$;

end

1.3 Data analysis and preprocessing

The data of this study is from https://www.drivendata.org/competitions/66/flu-shot-learning/page/213/. The National 2009 H1N1 Flu Survey (NHFS) was sponsored by the National Center for Immunization and Respiratory Diseases (NCIRD) and conducted jointly by NCIRD and the National Center for Health Statistics (NCHS), Centers for Disease Control and Prevention (CDC). The NHFS was a list-assisted random-digit-dialing telephone survey of households, designed to monitor influenza immunization coverage in the 2009-10 season. The target population for the NHFS was all persons 6 months or older living in the United States at the time of the interview. Data from the NHFS were used to produce timely estimates of vaccination coverage rates for both the monovalent pH1N1 and trivalent seasonal influenza vaccines.

This data has two target variables:

- **h1n1_vaccine**: Whether the respondent has received the H1N1 influenza vaccine.
- **seasonal_vaccine**: Whether the respondent received the seasonal influenza vaccine.

Both are binary variables: 0 = no; 1 = yes. Some respondents had not received either vaccine, others had received only one vaccine, and some had received both.

Table 1: Label Example

respondent_id	0
Field	Value
h1n1_vaccine	0
seasonal_vaccine	0

The data also contains 35 features as shown in Table.2. More details and descriptions of the features could be checked from the above website.

Table 2: Data Example

respondent_id	0
Field	Value
h1n1_concern	1
h1n1_knowledge	0
behavioral_antiviral_meds	0
behavioral_avoidance	0
behavioral_face_mask	0
behavioral_wash_hands	0
behavioral_large_gatherings	0
behavioral_outside_home	1
behavioral_touch_face	1
doctor_recc_h1n1	0
doctor_recc_seasonal	0
chronic_med_condition	0
child_under_6_months	0
health_worker	0
health_insurance	1
opinion_h1n1_vacc_effective	3
opinion_h1n1_risk	1
opinion_h1n1_sick_from_vacc	2
opinion_seas_vacc_effective	2
opinion_seas_risk	1
opinion_seas_sick_from_vacc	2
age_group	55 - 64 Years
education	< 12 Years
race	White
sex	Female
income_poverty	Below Poverty
marital_status	Not Married
rent_or_own	Own
employment_status	Not in Labor Force
hhs_geo_region	oxchjgsf
census_msa	Non-MSA
household_adults	0
household_children	0
employment_industry	-
employment_occupation	-

From this example, it could be noticed that there are some null values and the character data needs to be encoded. The feature after encoder could be checked in Table.3.

Table 3: Data Example

respondent_id	1
Field	Value
h1n1_concern	3
h1n1_knowledge	2
behavioral_antiviral_meds	0
behavioral_avoidance	1
behavioral_face_mask	0
behavioral_wash_hands	1
behavioral_large_gatherings	0
behavioral_outside_home	1
behavioral_touch_face	1
doctor_recc_h1n1	0
doctor_recc_seasonal	0
chronic_med_condition	0
child_under_6_months	0
health_worker	0
health_insurance	0
opinion_h1n1_vacc_effective	5
opinion_h1n1_risk	4
opinion_h1n1_sick_from_vacc	4
opinion_seas_vacc_effective	4
opinion_seas_risk	2
opinion_seas_sick_from_vacc	4
age_group	1
education	0
race	3
sex	1
income_poverty	2
marital_status	1
rent_or_own	1
hhs_geo_region	1
census_msa	0
household_adults	0
household_children	0
employment_industry	12
employment_occupation	19

Data normalization is a preprocessing step that is often performed on data before it is used for machine learning. The goal of normalization is to scale the data so that it has a consistent range and mean, which can help improve the performance and stability of machine learning models. The formula for data normalization is:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

where x is a data point, μ is the mean of the dataset, σ is the standard deviation of the dataset, and x_{norm} is the normalized data point.However, in the course of the experiment, it was found that the normalization of the data for this study did not have a great impact on the accuracy rate, which will not be discussed

too much here.After pre-processing the data, we obtained 6437 data for training, and then we divided the data in a ratio of 7:3 for the training set to the test set and put it into the random forest model.

2 Modeling and analysis

This training took place on a CPU-12700, unlike complex deep learning models that require a GPU, the random forest model in this case can also be run quickly on a CPU, which is one of the reasons we chose this model. The experiments are based on a *python* architecture and mainly use *scikit-learn*, a package about machine learning. The specific code can be viewed in the attached *jupyter notebook* file. During the training process, there exist some hyperparameters that need to be set manually and can not be learned by the model itself.

- *n_estimators*: This hyperparameter controls the number of decision trees that are used in the random forest. Increasing the number of trees can improve the accuracy of the model, but may also increase the training time and memory usage. During the experiment, the model could get an excellent performance by setting it to *10*.

- *criterion*: This hyperparameter specifies the function used to measure the quality of the split at each node in the decision trees. In a random forest, each tree is constructed using a random subset of the training data and features. For binary classification problems, we choose *entropy* as the metric. The entropy criterion is used to measure the impurity of a split, where a split is considered good if it reduces the impurity of the child nodes.

- *min_samples_leaf*: This hyperparameter controls the minimum number of samples required to be at a leaf node in the decision trees. A smaller value of *min_samples_leaf* can result in more complex trees that may overfit the training data, while a larger value can result in simpler trees that may underfit the data. We choose it as *3*

- *max_depth*: This hyperparameter controls the maximum depth of each decision tree. A smaller value of max_depth can help prevent overfitting by limiting the complexity of the trees, while a larger value can result in more complex trees that may overfit the training data. We choose the maximum depth of each decision tree as *10*.

Because there are two different labels in this experiment and we do not see them as a multi-classification problem but as two binary problems, we have to create two different random forests. A part of the decision tree in the random forest is shown in the Fig.1. (Cause the whole decision tree is too large, more details could be checked in the attached files.)

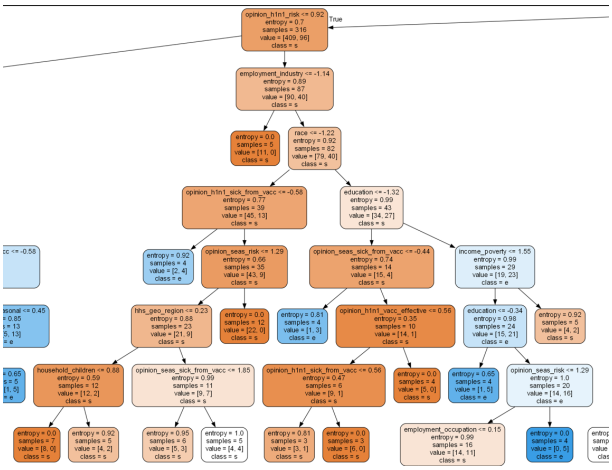

Figure 1: The part of one decision tree

And the accuracy of the RF model of two labels is shown in Table.4.

	Accuracy Score
H1N1	0.809
Seasonal	0.794

Table 4: Accuracy Scores for H1N1 and Seasonal Datasets

The results of the model are acceptable in terms of accuracy, and for further analysis we can analyse the confusion matrix of the model

A confusion matrix is a table that is often used to evaluate the performance of a machine learning model. It compares the predicted values of the model with the actual values, and shows how many true positives, true negatives, false positives, and false negatives the model produced.

The importance of the confusion matrix lies in its ability to provide a more detailed and nuanced understanding of the performance of a machine learning model than a single metric such as accuracy or precision.

For example, a model with high accuracy may still have poor performance on specific classes or instances within the dataset. A confusion matrix can help identify these areas of poor performance and provide insights into how to improve the model.

Some of the key insights that can be gained from a confusion matrix include:

- **True positives**: The number of instances that the model correctly predicted as positive.
- **True negatives**: The number of instances that the model correctly predicted as negative.

- **False positives**: The number of instances that the model incorrectly predicted as positive.
- **False negatives**: The number of instances that the model incorrectly predicted as negative.
- **Precision**: The proportion of positive predictions that are actually true positives.
- **Recall**: The proportion of true positives that are correctly identified by the model.
- **F1 score**: A weighted average of precision and recall, which provides a balanced evaluation of the model's performance.

By examining the values in a confusion matrix, we can gain a better understanding of the strengths and weaknesses of the model, identify areas for improvement, and make more informed decisions about how to optimize the model for better performance. The confusion matrixes of the two models are shown below respectively:

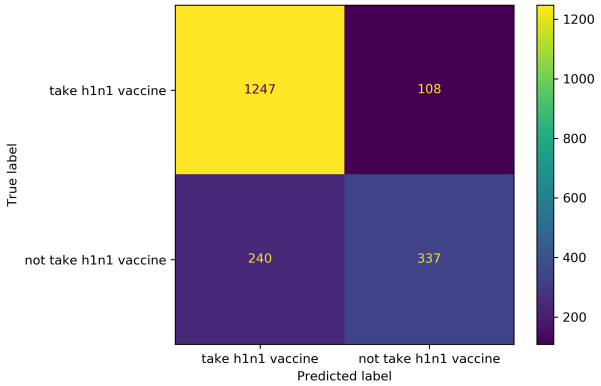

Figure 2: The confusion matrix of H1N1 vaccine dataset

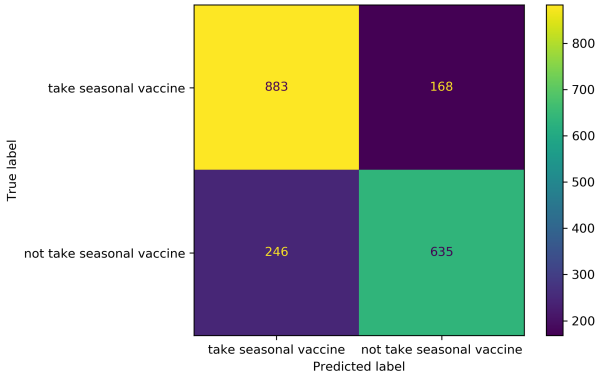

Figure 3: The confusion matrix of seasonal vaccine dataset

The confusion matrix for the seasonal model shows that out of a total of 1932 instances, the model correctly predicted 1518 (883 true negatives and 635 true positives) and incorrectly predicted 414 (246 false negatives and 168 false positives). The model's precision can be calculated as 635 / (635 + 168) = 0.791 and its recall can be calculated as 635 / (635 + 246) = 0.720. These values indicate that the model correctly identifies a high proportion of positive instances (recall), but also produces a relatively high number of false positives.

The confusion matrix for the h1n1 model shows that out of a total of 1932 instances, the model correctly predicted 1584 (1247 true negatives and 337 true positives) and incorrectly predicted 348 (240 false negatives and 108 false positives). The model's precision can be calculated as 337 / (337 + 108) = 0.757 and its recall can be calculated as 337 / (337 + 240) = 0.584. These values indicate that the model correctly identifies a lower proportion of positive instances than the seasonal model (lower recall), but produces fewer false positives.

Overall, these analyses suggest that the h1n1 model performs slightly better than the seasonal model, although both models produce a relatively high number of false positives.

The next step is to analyse the effect of different features on the importance of the outcome of whether or not to vaccinate. We could use the Gini index to deal with this problem. The Gini index, also known as the Gini impurity or Gini coefficient, is a measure of impurity used in the context of decision trees and random forests. It measures the degree or probability of a particular feature being classified incorrectly when it is randomly chosen. In other words, it measures the likelihood of an item being misclassified into the wrong class by a decision tree model.

The Gini index is calculated as follows:

$$G_i = 1 - \sum_{j=1}^{k} p_{i,j}^2$$

where G_i is the Gini index for node i, k is the number of classes, and $p_{i,j}$ is the proportion of instances of class j in node i.

We denote variable importance measures as VIM and Gini index as G_i. Suppose there are J features $X_1, X_2, X_3, ..., X_J$, I decision trees, and C classes. We want to calculate the Gini index scores $VIMj(Gini)$ for each feature X_j, which is the average change in impurity of nodes splitting on feature X_j over all decision trees.

The VIM score of feature X_j at node q in the i-th decision tree, which is the change in the Gini index before and after splitting on X_j, is calculated as:

$$VIM_{jq}^{(Gini)(i)} = GI_q^{(i)} - GI_l^{(i)} - GI_r^{(i)}$$

If feature X_j appears in the set of nodes Q in decision tree i, then the VIM

score of feature X_j in the i-th decision tree is:

$$VIM_j^{(Gini)(i)} = \sum_{q \in Q} VIM_{jq}^{(Gini)(i)}$$

Assume that there are I trees in RF, then:

$$VIM_j^{(Gini)} = \sum_{i=1}^{I} VIM_j^{(Gini)(i)}$$

Finally, we can normalize the VIM scores for all features X_j as follows:

$$VIM_j^{(Gini)} = \frac{VIM_j^{(Gini)}}{\sum_{j'=1}^{J} VIM_{j'}^{(Gini)}}$$

We plot the final feature importance results as a bar chart as shown below:

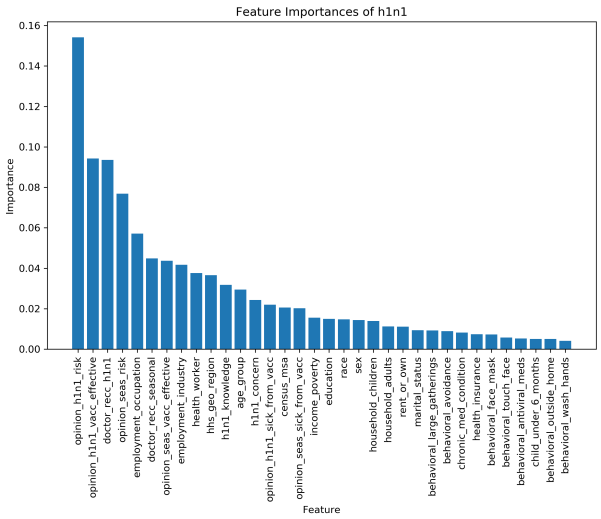

Figure 4: The importance bar chart of h1n1 vaccine dataset

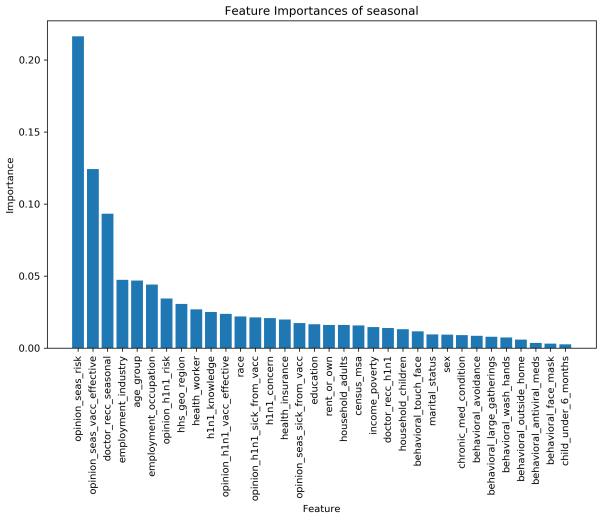

Figure 5: The importance bar chart of seasonal vaccine dataset

Based on the feature importance scores of the h1n1 dataset, we can see that the top three most important features in predicting h1n1 vaccination are: *opinion_h1n1_risk*, *opinion_h1n1_vacc_effective*, *doctor_recc_h1n1*. The remaining features have a smaller impact on the prediction, with the least important feature being *behavioral_wash_hands*. These results suggest that people's perceived risk of getting infected with h1n1, their beliefs about the effectiveness of the h1n1 vaccine, and the recommendation of the vaccine by doctors are the most important factors in determining whether they get vaccinated or not.

Based on the feature importance scores of the seasonal dataset, we can see that the top three most important features in predicting seasonal vaccination are: *opinion_seas_risk*, *opinion_seas_vacc_effective*, *doctor_recc_seasonal*. Similar to the h1n1 dataset, the importance of these features suggests that people's perceived risk of getting infected with seasonal flu, their beliefs about the effectiveness of the seasonal flu vaccine, and the recommendation of the vaccine by doctors are the most important factors in determining whether they get vaccinated or not. The remaining features have a smaller impact on the prediction, with the least important feature being *child_under_6_months*. It is worth noting that some features appear to have a higher impact on the prediction for the seasonal dataset than they did for the h1n1 dataset, such as *opinion_seas_risk* and *opinion_seas_vacc_effective*.

Overall, the feature importance scores provide insights into which features are most important in predicting vaccination for each dataset and provide strong supporting data for our thematic research.

3 Conclusion

In this topic, a Random Forest prediction model was developed to forecast the uptake of H1N1 and seasonal influenza vaccines using individual and collective

behaviors. The model was trained and tested on two separate datasets: one for H1N1 vaccine uptake and the other for seasonal vaccine uptake. More details of the implementation could be checked in the attached *Data Analysis.pdf*.

The development of a Random Forest prediction model for forecasting H1N1 influenza vaccine and seasonal influenza vaccine uptake using individual and collective behaviors is an important step towards improving public health outcomes. Influenza is a highly contagious respiratory illness that can cause severe illness and even death in vulnerable populations, and vaccination is one of the most effective ways to prevent the spread of the virus.

The study showed that the Random Forest model was able to accurately predict vaccine uptake for both H1N1 and seasonal influenza, with accuracy scores of 0.81 and 0.79, respectively. This suggests that the model can be a valuable tool for healthcare professionals and public health officials in predicting vaccine uptake rates and targeting interventions to improve vaccination rates.

The feature importance scores provided valuable insights into which factors were most important in predicting vaccine uptake for each dataset. For H1N1 vaccine uptake, the most important features were opinion on H1N1 risk, opinion on H1N1 vaccine effectiveness, and doctor recommendation for H1N1 vaccine. For seasonal influenza vaccine uptake, the most important features were opinion on seasonal influenza risk, opinion on seasonal influenza vaccine effectiveness, and doctor recommendation for seasonal influenza vaccine.

Future work in this area could involve expanding the study to include other influenza strains or vaccines, such as the COVID-19 vaccine. Additionally, the model could be refined or improved through the use of alternative algorithms or techniques, such as deep learning or ensemble methods, to further improve accuracy and predictive power. Finally, the insights gained from this study could be used to inform public health campaigns or interventions aimed at improving vaccine uptake rates, particularly in vulnerable populations or regions with lower vaccination rates.