MACHINE LEARNING ON PREDICTION OF H1N1 FLU VACCINATION STATUS OF AN INDIVIDUAL

Team member	Student #	Task Completed	% of the Project
Jinlun Zhang	20055835	- Responsible for most of the code works such as data pre-processing (convert all the features into numeric, implement the count rank encoding strategy), data cleaning (Filling the missing values), EDA (correlation heatmap, missing value graph), training of the baseline model (SVM) and implementing the stacking ensembling method - Responsible for the report writing and the slides editing corresponding to the codes I am responsible of	60
Wenqi Tang	20093622	- Responsible for some of data pre-processing, EDA, and the training of the MLP model - Responsible for the report writing and the slides editing corresponding to the codes I am responsible of	20
Eissa Khan	20082302	- Responsible for coding and training of the XGBoosting model - Responsible for partial data pre-processing by implementing the over sampling strategy - Responsible for report writing and slide editing corresponding to the codes I am responsible of	20

Used Software: Google Colab, VSCode, Jupyter Notebook

Used Package (Python): Pandas, Matplotlib, Seaborn, Sklearn

Problem Definition

As a key public health measure to fight against infectious diseases, vaccines provide the immunization for individuals and adequate immunization in a community that can further reduce the spread of diseases through "herd immunity". As the world attentions have all been directed to the COVID-19 vaccines, we are thinking that it may be worth-while to revisit the public health responses to a different recent major respiratory disease pandemic (H1N1) to see if we can reveal some correlations or similarities between the 2 pandemics and determine certain peculiarities from the comparison.

Beginning in Spring 2009, a pandemic caused by the H1N1 influenza virus swept across the world. A vaccine for the H1N1 flu virus soon became publicly available in October 2009. In late 2009 and early 2010, the US conducted the National H1N1 Flu Survey, asking respondents whether they had received the H1N1 flu vaccines or not, in conjunction with questions about themselves. These additional questions covered their social, economic, and demographic background, opinions on risks of illness and vaccine effectiveness, and behaviors towards mitigating transmission. This is the source of the dataset we will use for this machine learning project to thoroughly understand how these personal characteristics or features are associated with the personal vaccination statuses to retrieve insights for the future public health efforts.

Therefore, the problem this paper attempts to address can be summarized to using machine learning strategies to predict on whether an individual has received a H1N1 flu vaccine or not based on his/her collected personal characteristics, which can be considered as a binary classification machine learning task.

Description of Data Exploration and Preparation

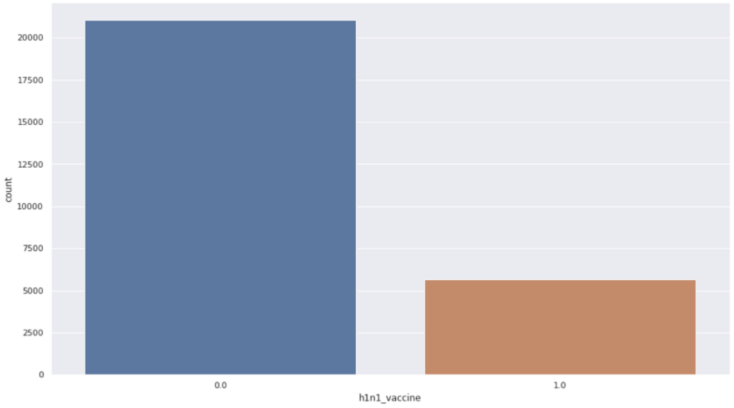

Figure 1: Count of the records by their target attribute values

We start the data exploration by checking the target attribute values and discover that this binary target attribute is moderately imbalanced as shown in Figure 1, indicating that the ideal model evaluation metrics we should choose for our model should be F1 score or AUC score or both. Moreover, to address this imbalance, we will employ the oversampling strategy in the data-preprocessing section to balance the imbalanced number of records in each of the target class.

Then, by exploring the numeric attributes values in the given dataset, we realize that all the provided numeric features are either binary numeric data (like the feature "behavioral_touch_face" where 1 indicates that a person has avoided touching his/her face, and 0 indicates that a person has not avoided) or categorical ordinal numeric data (like the feature "opinion_h1n1_vacc_effective"), meaning that the numeric feature values are discrete, and their maximal values are below or equal to 5 as shown in Figure 2.1. As a result, no numerical data transformation like scaling may be necessary for this dataset, because the scale of each numeric feature is similar to each other, while we want to preserve the ordering within the categorical ordinal numeric features.

Furthermore, through observing the correlation matrix of all the numeric features as shown in Figure 2.2, we can conclude that most of the pairs of the numeric features are positively correlated, and those numeric features are also positively correlated with the target attribute, where some of the numeric features are highly correlated with the target feature such as the feature ‘doctor_recc_h1n1’ with correlation being 0.4 and the feature ‘opinion_h1n1_risk’ with correlation being 0.3. Thus, we may expect to achieve a high lower bound on the model evaluation score for this task (For example, our baseline model should produce at least a evaluation score more better than 50%)

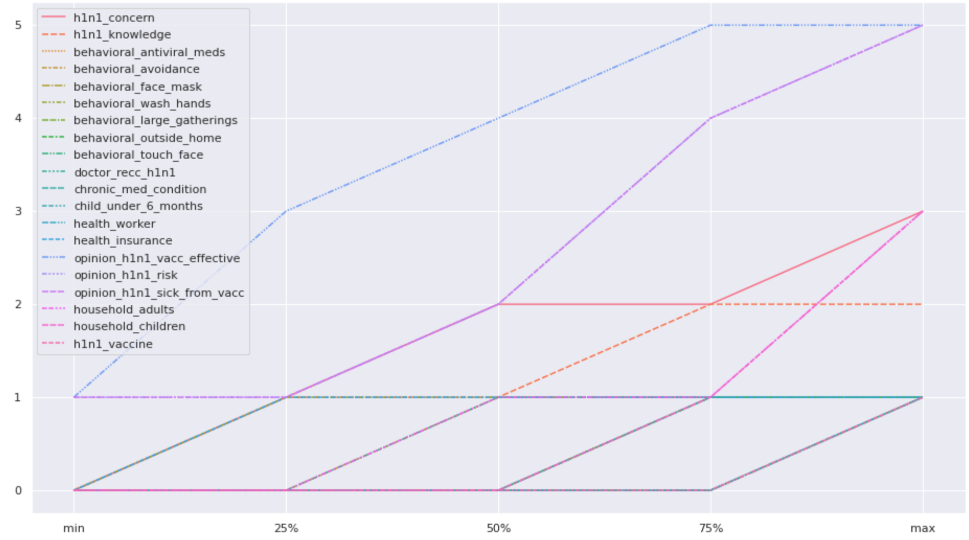

Figure 2.1: Distribution of the numeric feature values

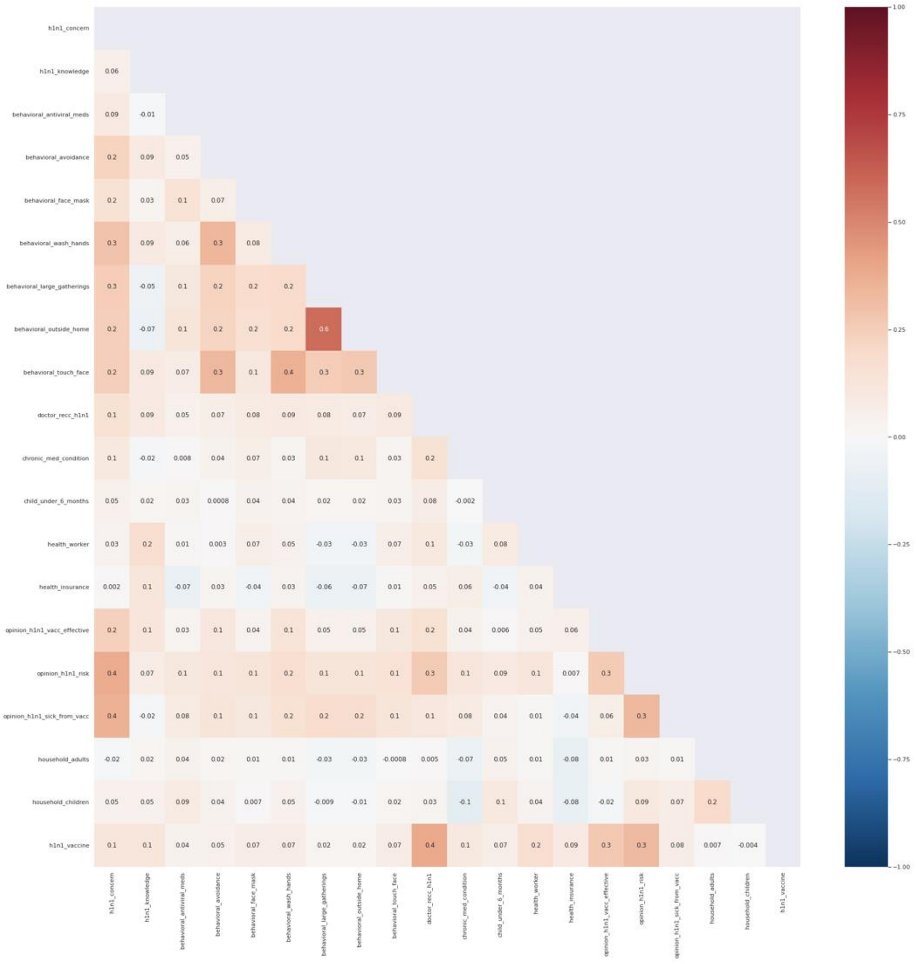

Figure 2.2: Headmap of the correlation matrix of all the numeric features, including the target feature 'h1n1_vaccine'

Moving to the exploration of categorical features, we start by inspecting the "age_group" feature, which has 5 unique categorical values with each indicating a range of age, implying that we can convert this categorical feature into numeric by replacing those categorical age ranges with the middle value of each age range as shown in Figure 3. It can also be observed from Figure 3 that the difference between the number of records in one age group and the number of records in another age group is about 1000, indicating that this age feature provides moderate amount of information for the model to learn (as no age group dominates the feature). In contrast, the "race" feature may provide insufficient amount of information to train the model as shown in the Figure 4, where the feature value 'White' dominates. Thus, to maximize the information extracted from the feature "race" and convert the feature values into numeric, we need to select a conversion strategy wisely. For example, we could choose to use the ordinal encoding strategy to assign a unique integer value to each of the "race" value (which will undesirably introduce intrinsic ordering to the "race" values), or we can bin the rest of 3 "race" values into one group (converting the 'race' feature into a binary feature with a certain degree of information loss). Hence, converting the 'race' feature into numeric with maximal data originality preserved can be considered as one challenge in tackling this machine learning task, and we will discuss our solution to this more in depth in the

Methodology section of the paper.

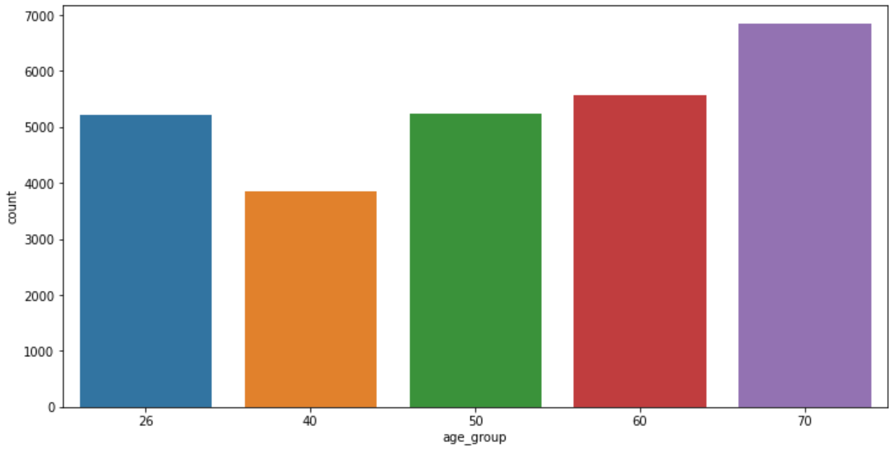

Figure 3: Distribution of “age_group” feature values after transformation

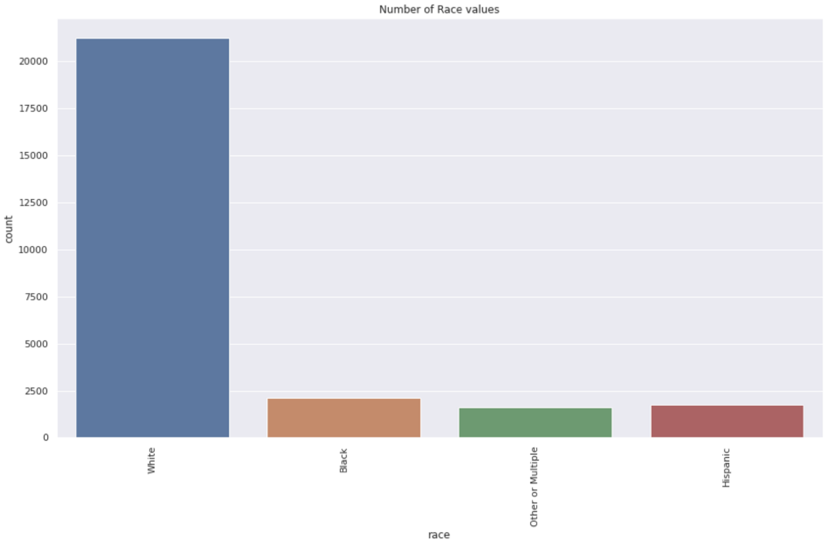

Figure 4: Distribution of “race” feature values

Regarding to the exploration of the missing values, as shown in Figure 5, approximately 14000 records have at least one feature value being missing, implying that the strategy of dropping records with missing values is not feasible in this situation where we will exclude nearly 50% of training data by doing so. In addition, through checking the proportion of the missing values in each feature of the dataset, we have discovered that there are 3 features (‘health_insurance’, ‘employment_industry’, ‘employment_occupation’) with around 50% of values being missing, and it is not advisable to drop the 3 features as they may contain crucial information for the model to accurately predict on the individual’s vaccination status. For example, a nurse who do not want to receive the vaccine may not want to disclose her career (so missing value in ‘employment_industry’ and ‘employment_occupation’), because the healthcare employees are usually required by the public to receive the vaccine. Therefore, in such case of “missing depending on the missing value itself”, those 3 features may provide abundant key information for model to produce more accurate predictions once their missing values have been filled appropriately.

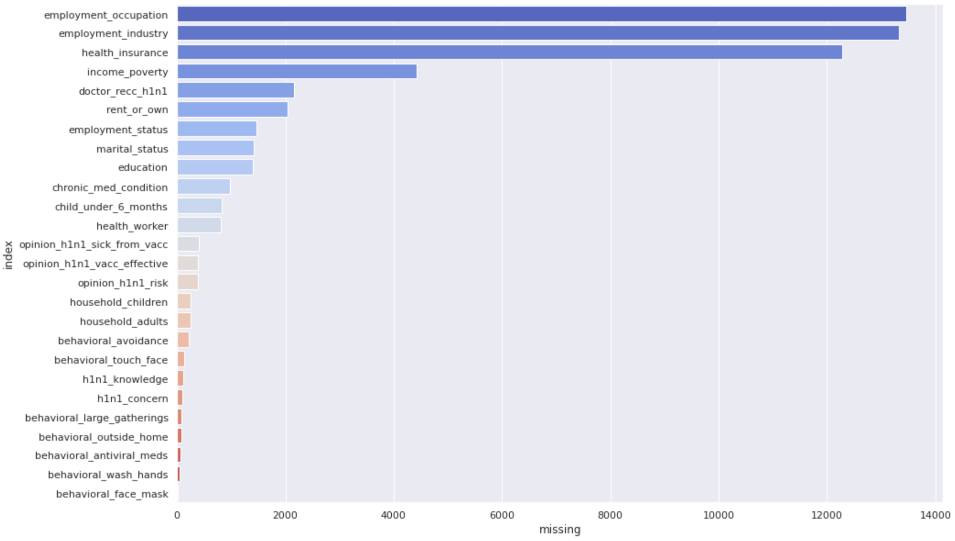

Figure 5: Count of missing values in features

Methodology

To fill in all the missing values in the dataset, the general statistic replacement (replacing missing values by a statistic associated with the column) is not considered as an acceptable approach, because it will lead to overrepresented values in features, especially in those features with around 50% of values being missing. Hence, to fill in the missing values with maximized variances in the features, a more reasonable strategy to use is imputation, namely using the k nearest neighbour (KNN) model to predict the missing values of a record based on the k most similar records to the one record with missing values.

To implement the imputation strategy, we need to convert all the categorical features into numeric in the data pre-processing stage. As for the categorical features that do not have intrinsic ordering in the values, such as the aforementioned feature “race”, we can use the one-hot encoding strategy to transform those categorical features into numeric with minimal information loss. However, we should restrict the range of application of the one-hot encoding to those categorical features that have less than 10 unique feature values. The reason of having such restriction is to avoid the dimensionality explosion that can lead to long training time and high possibility of overfitting. Thus, for the categorical features that do not have intrinsic ordering and have more than 10 unique feature values, such as the feature “hhs_geo_region”, we will use the count rank encoding strategy instead (replacing a categorical value by the rank of its frequency in the feature), which is empirically proved to be a reliable conversion strategy for the categorical features without intrinsic ordering. Then, for the categorical features with intrinsic ordering, such as the feature “education”, we can utilize the ordinal encoding strategy to convert them into numeric with their ordering reflected in the numerical representation.

After transforming all the categorical features into numeric, we can apply the KNN imputation strategy to handle all the missing values in the dataset. However, before proceeding to the model training section, we need to implement the oversampling strategy as the target attribute ('h1n1_vacine') is heavily imbalanced with the vaccinated group containing less than 25% of the data records. The data is oversampled by using random oversampling. Effect of random oversampling resulted in new rows being added to the data by choosing random rows from the vaccinated group (with replacement) until the two groups have equal number of rows. Thus, we end up with a training set that has equal number of data records in each of the target class.

Moving to the model training section, the first model we decide to implement is the Support Vector Machine (SVM), serving as the baseline model. The incentives of choosing SVM as the baseline model include that the SVM has fewer hyper-parameters for tuning (quickly converge to an optimal hyper-parameter solution state), simple to implement, and requires less training time to give an impression about the complexity of the classification task. In addition of being a baseline model, SVM is also considered to be a qualified candidate model for solving this binary classification task, because SVM is capable of producing adequate generalizability for binary classification task due to the identified optimal hyperplane (may even outperforms neural networks occasionally), and the Kernel functions can also be exploited to increase the feature space and cause the dataset sparser to yield a satisfactory model performance.

With the utilization of random search CV (determining a possibly satisfactory set of hyper-parameters by testing multiple random combinations out of the predefined ranges of hyper-parameter values), the final validation AUC and F1 scores the SVM achieved are 75.7% and 75.6%. Additionally, it is noteworthy that the validation AUC and F1 scores the SVM achieved are only 78% and 50% when the count encoding strategy is applied, meaning that we improve the model's AUC score dramatically by merely using the count rank encoding strategy rather than the count encoding strategy, which reflects the importance of empiricism in machine learning.

Furthermore, we choose to use Multi-Layer Perceptron (MLP) as the second model to implement because it is suitable for tabular data set, while non-linear models can be learned in real time. We import MLP Classifier and run the model on the validation set to evaluate how it performs and use the validation result as the feedback for hyper-parameter tuning. Eventually, with the help of random search CV method, we reach the locally optimal hyper-parameter setting and the optimal NN architecture design by configuring 256 hidden nodes in the first hidden layer, 128 nodes in the second hidden layer, and 64 nodes in the third hidden layer of the MLP model. As a result, the MLP yields 75.8% for both final validation AUC and F1 scores, which only outperforms the baseline model (SVM) for 0.1%. A possible explanation for such inappreciable outperformance may be that the MLP is not a feasible model for extracting underlying patterns from the training dataset as the data records may not be adequately sparse, while the SVM can use Kernel functions to project the records into higher dimensional space to disperse the data records and classify them, leading SVM to have a comparable performance with MLP. Additionally, according to the 'No Free Lunch' theorem, it is not sensible to assume that the deep learning model will always outperform the relatively less complicated statistical model for any given task, meaning that the MLP does not perform well due to the nature of the problem (or the infeasibility of the model itself for the task).

As for the third model we choose to implement, XgBoost is selected because this model is typically the best for tabular data with mixed data types (nominal and categorical). XgBoost is also an improvement on model method decision trees, which are constructed greedily and a single tree is constructed with the purpose of fitting the training data. XgBoost is an ensemble technique that iteratively constructs decision trees on the residuals in errors from prior trees. For hyperparameter search, random search CV is performed and evaluated against cross validation, leading to the conclusion that this model (XgBoost) performed very well overall with validation AUC and F1 scores averaging above 91%.

Finally, a stacking ensembling strategy is implemented to combine the 3 models we trained together to produce more educated final predictions. To achieve even further robust prediction results, 2 additional models, Random Forest and LightBGM, are added in the stacking in attempts to diversify the base learners (and so obtain better generalizability). As a result, both of the final validation AUC and F1 scores the stacking ensembling strategy achieved are 95.2%, indicating sufficiently accurate predictions produced by the stacking ensembling strategy.

	Validation AUC	**Validation F1**
SVM	75.7%	75.6%
MLP	75.8%	75.8%
Xgboost	91.7%	91.7%
Stacking Ensembling	95.2%	95.2%

Table 1: Summary of the evaluation metrics scores gained by the 3 trained models and the 1 stacking ensembling method evaluated on the validation dataset.

Conclusion and Future Works

In conclusion, through analyzing the non-sensitive personal characteristics information, a machine learning solution can be formulated to effectively and accurately predict on the vaccination status of an individual.

Nevertheless, the solution proposed in this paper may not be optimal, because there are substantial future works to further enhance the effectiveness and the reliability of the solution. For example, instead of using the KNN model to impute the missing values, the more sophisticated deep learning model can be utilized to complete the imputation task, which may further strengthen the model performance and does not necessitate the input features to be numeric (as the Feature Encoder in the deep learning model can handle categorical data). Moreover, since the original dataset only contains approximately 27000 records, it is desirable to conduct more researches online to integrate some external data to enlarge the existing training dataset to acquire more variances for the model training, which can further boost the reliability of this machine learning solution. Last but not least, it is also sound to execute multiple ensembling strategies, such as Bootstrap Aggregating (Bagging) and Boosting, to identify the optimal ensembling strategy for this specific task and to use some regularization methods to avoid potential overfitting of the ensembling strategy.