Core Project
(PRJ3001)

Final Report

Flu Shot

By

Edapalapati Venkata Abhiram

1800152C203

Sangu Pranav

1800256C203

Bollavaram Golla Riteesh Ram Chander

1800206C203

Supervisor

Dr. Sohrab Hossain

Associate Professor

Department of Computer Science and Engineering
School of Engineering and Technology

December, 2021

Acknowledgement

We would like to express my sincere gratitude to my supervisor Dr. Sohrab Hossain, our mentor who gave us the golden opportunity to do this wonderful project on the topic of Flu Shot Vaccination Program for giving us all the support required and clarifying all our doubts throughout the making of the semester core project as well as for giving us the freedom to take our work in the direction we desired. His guidance and advice carried me through all the stages of writing my project. I would also like to thank my teammates and friends for helping me and encouraging me towards the completion of the project.

Thanking You.

Edapalapati Venkata Abhiram 1800152C203

Sangu Pranav 1800256C203

Bollavaram Golla Riteesh Ram Chander 1800206C203

Computer Science and Engineering

December 4, 2021

BML Munjal University, Gurgaon, Haryana

CANDIDATE'S DECLARATION

We, Edapalapati Venkata Abhiram, Sangu Pranav, Bollavaram Golla Riteesh Ram Chander, hereby declare that the work done in my core project entitled *"Flu Shot*" in fulfillment of completion of 7th semester of Bachelor of Technology (B-Tech) program in the Department of Computer Science and Engineering, BML Munjal University is an authentic record of our original work carried out under the guidance of Dr. Sohrab Hossain, Associate Professor.

Due acknowledgements have been made in the text of the project to all other materials used.

This core project was done in the full compliance with the requirements and constraints of the prescribed curriculum.

Edapalapati Venkata Abhiram	1800152C203	E. Abhi Ram
Sangu Pranav	1800256C203	S. Pranav
Bollavaram Golla Riteesh Ram Chander	1800206C203	B. G. Riteesh

Place: BML Munjal University

Date: December 4, 2021

CERTIFICATE

This is to certify that the Core Project entitled “Flu Shot” to the best of my knowledge is a record of the bona fide work carried out by Mr./Ms. Edapalapati Venkata Abhiram, Sangu Pranav, Bollavaram Golla Riteesh Ram Chander under my guidance and/or supervision. The contents embodied in this report, to the best of my knowledge, have not been submitted anywhere else in any form for the award of any other degree or diploma. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried out during *July - December 2021* as part of their 7th semester coursework for Bachelor of Technology (B-Tech) program in the Department of Computer Science and Engineering, BML Munjal University.

Name and Designation of the Supervisor:

Signature:

Date:

Place:

Contents

Abstract

Vaccination plays a vital part in public health care and is one of the most time-consuming processes on the planet. In this study, we look into how machine learning approaches may be used to predict vaccination willingness. Artificial Intelligence's goal is to make computers more effective in tackling difficult healthcare problems, and we may utilize computers to understand data obtained from various surveys. It aids in the early discovery and resolution of numerous issues that impact immunization rates. Computer-assisted diagnosis, decision support systems, expert systems, and software implementation have all demonstrated their value in health care, with various algorithms reducing mistakes and controlling disease development. " The study focuses on the 2009 human flu outbreak, sometimes known as "swine influenza" or H1N1 influenza, in which vaccination has proven challenging. Data mining and machine learning approaches, for example, are non-clinical tools that might aid with this. Using Machine Learning Classification Algorithms, this study will examine the H1N1 Vaccination Progress in the United States. The paper's conclusions reveal which method is best for a taken dataset. The suggested infrastructure in this study can assist vaccine providers by giving external Machine Learning support.

1. Problem Definition:

Influenza (flu) is a potentially fatal disease that can result in hospitalizations and, in rare cases, fatality. Despite the fact that every flu season is different from other and affect differs from person to person , yet crores of human beings catch the flu each year, lakhs are sick, and thousands of people die from flu-related . The flu can cause a few days of feeling unwell and missing work, or it might cause more serious diseases. sinus infections, ear infections, Bacterial pneumonia, and aggravation of existing health problems such as asthma, congestive heart failure, or diabetes can all be complications of the flu. The best method to help guard against influenza is to get a seasonal flu vaccination every year.

Vaccination has been demonstrated to provide a variety of benefits, namely, hospitalizations, mitigating the risk of flu infections, and sometimes even influenza - related fatality among child's, as well as lowering the intensity of disease. It is difficult to identify how the each citizen is acting and willing to take their H1N1 and seasonal flu vaccines, so we are going to predict whether a person got vaccinated or not based on the information shared by him like his health behaviors, opinions, and background which helps in government to stimulate towards the vaccination before it's late.

2. Project Objectives:

Since the vaccines will decrease the chance of getting affected by Influenza or seasonal flu diseases. Taking those vaccines will increase public health and prevents deaths and financial losses.

a) Create a Unique platform which will help in finding the people who are suspected to not to take the vaccine
b) Make the all-possible features in to consideration for predicting the person's probability like demographic background, opinions on flu and vaccines, their social and economic backgrounds.
c) Spread the awareness to the targeted people who are suspected of not to take.
d) Stop the Spread of disease after the vaccine made available.
e) Reliable, cost efficient and user-friendly software.

In 2009 the US government conducted an H1N1 flu telephonic survey, in that survey they gathered the information from each member on whether they took their seasonal flu vaccines and H1N1 vaccines or not due to lack of Intelligence on the people behavior and the data. This can be solved by predicting the individual and the government can take appropriate action to make people vaccinated.

3. Challenges:

One of the basic challenges for the any of the Artificial Intelligence and Machine Learning projects is the finding/ gathering of the required data. Another importance aspect is the accuracy of the data collected and filtration of the data which generally contains the missing values and zeros irrespective of the field. The data challenges contain we faced in this project are

- Inadequate training data
- Information of low quality.
- Features that aren't relevant.

- Training data that really isn't appropriate

Model selection is one of the biggest challenges for the prediction of the as the different models behave differently on the same training data. The training needs to be monitored carefully else it leads into the overfitting and under fitting of the model.

These challenges mainly solved by

1. Select a best model, one with the relevant parameters.
2. Selection of the number of attributes in training data.
3. Reduce the noise.
4. Constraining the model.

Every application needs to have some specific GUI as it helps the user to understand easily and gets to know how to use it. Designing the interface is the challenging part for the officials which helps in identifying the individual. Since the data includes the public health it contains many regulations like IT regulations, compliances, standards and data where and how it is used.

4. Deliverables:

a) The resultant output will contain 3 columns like respondent_id, h1n1_vaccine, seasonal_vaccine.
b) Two target elements should be predicted with float probability ranging from 0.0 to 1.0.
c) These probabilities should be useful in identifying the persons who are more suspected of not taking vaccines can be also vaccinated against flu and can stop the spread of the flu.
d) A responsive UI which can predict the given data.

5. Literature Review:

Swine flu influenza H1N1:

H1N1 is a Swine flu influenza virus that causes illness to the respiratory system of a person. This virus is infectious and is spread by inhalation or ingestion of contaminated droplets, such as those produced by sneezing or coughing. This H1N1 virus is made up of many pieces, including genetic material that allows the virus to replicate itself. The virus's outer surface is coated with protein spikes known as H and N proteins. The respiratory system cells like cells of Lungs, Throat and Nose were attacked by the Swine flu. The virus enters into the cell body by binding it's H spike to the receptors of the cell. Now the virus takes path towards cytoplasm of a cell and releases virus genetic info into it. The virus now travels to the cell's nucleus and opens up to release its genetic information into the cytoplasm. As the virus genetic info reaches into the nucleus and now the nucleus will use all its machinery to create duplicate genetic info of the virus. Few of those copies return to the cytoplasm, where they bind to ribosomes. Now, depending on the copies, the ribosomes produce proteins such as H and N spikes, which are required to construct a new virus molecule. All of the molecules will combine below the cell membrane to form a new virus, allowing a single cell to generate millions of viruses. These newly generated viruses will burst the

cell membrane and will go on infecting other healthy cells causing the flu, this even leads to death of the infected person. To prevent or decrease the changes of death one should get vaccinated, the vaccine will prepare immune system for the battle with the virus by triggering it to make antibodies. Therefore, taking H1N1 vaccine for this flu will reduce the chances of risk by 50%. Hence getting vaccine is most important for this reason we are goanna to build a machine learning model which can predict the probabilities of a person getting vaccinated for both H1N1 and seasonal flu viruses. For that we have gone through training various machine learning algorithms/models for our available dataset of 26,000 respondents, the accuracy scores of those models were noted. Further in this section we will discuss about each and every model in detail, how we trained them, their disadvantages and recorded accuracy scores.

Logistic Regression:

This is generally used for binary classification problems and one can also use logistic regression for multiclass classifications. If one tries to classify dependent variable using linear regression then two problems occur mainly, if there were outliers in the dataset best fit line will be deviated and that leads to misclassification of target variable values and the second problem is that most of the times outputs will be greater than 1 and less than 0. Therefore, one should use logistic regression to overcome these issues. Here logistic regression uses sigmoid function which will quash the straight line. Now let's discuss the intuition behind logistic regression, logistic regression is usually applied to a problem statement where two classification problems can be linearly separable. To make the classes linearly separable one need to use cost function to estimate best partitions here, the logistic regression cost function will note all the positive and negative cases of each and very datapoint, the negative value indicates misclassified. Therefore, the cost function will make sure that the summation of all the points with its distances should be maximum. The cost function will be like:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

Here logistic regression will try to optimize the above equation but the problem arises when there were outliers, in that situation the distance from the negative outlier will be more and that leads to negative outcomes of the above equation. So, the algorithm may think that it's not the best fit line to separate the classes. Therefore, to overcome this problem we will apply sigmoid function for this equation. The final equation will become:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

Here after calculating every data point value that will be passed to the sigmoid function and sigmoid will make sure that the value is transformed between 0 to +1, so the effect of outlier will be removed and sigmoid will be acting as an activation function. The graph of sigmoid function is given below:

In this way logistic regression works internally and some of the disadvantages of using logistic regression is as follows:

- There will be a problem of overfitting if the number of features is more than the number of observations.
- It assumes linearity between independent variables and dependent variable.
- One can't solve nonlinear problems using logistic regression.
- Multicollinearity among independent variables should not be there while using logistic regression.

The recorded ROC-AUC score for logistic regression in our prediction is:

H1N1 Vaccine: ***0.831***

Seasonal Vaccine: ***0.856***

Support Vector Machine:

Support vector Machine can be used for both classification as well as regression problems, now let's understand about support vector machine. The main goal of SVM is to find a hyperplane in an N-dimensional spaces, where N stands for number of independent variables/features. In SVM there will be support vectors, these vectors are the data points that are closest to the hyperplane on both the sides and the position of the hyperplane is influenced by those support vectors. Like the algorithm draws a margin on the both sides of hyperplane and those margins will pass through support vectors. Now the algorithm will find the best fit hyper plane where the marginal distance is maximum, if the marginal distance is maximum, one will get more generalization model and that decreases error rate for prediction. Now what if the data is not linearly separable? Then the SVM uses a technique called as SVM kernels, these kernels will try to convert low dimension into a higher dimension so that the data points are linearly separable and we can draw hyperplanes and margins. SVM will consider two separate positive and negative classes and when the below given equation doesn't satisfy for any instance/datapoint then one can say that the datapoint was misclassified.

$$y_n\left[w^T\phi(x) + b\right] = \begin{cases} \geq 0 \; if \; correct \\ < 0 \; if \; incorrect \end{cases}$$

SVM has an idea of soft margin where the algorithm allows some misclassified data because if the hyperplane is too specific in terms of classification, then the model

may become overfitted now if the model see any noisy data or in case of test data the accuracy may reduce. Even though with soft margin we see some misclassified data but the model will be more generalized.

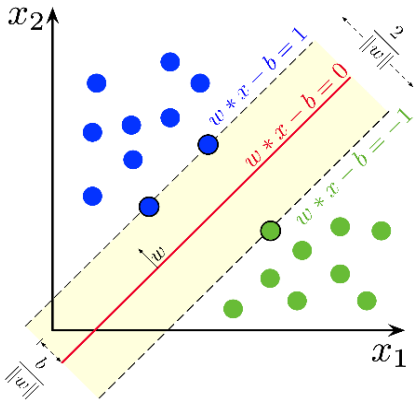
The optimization function for SVM is:

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

Here c and gamma are the hyperparameters, c will consider or gives penalty for every misclassified data. If c is high then penalty will be high for misclassified data, if that is high then margin lines with small margin will be considered. In case of gamma if its value is low then in order to be considered in same group the datapoints should be close with each other. For high values of gamma many datapoints were grouped together. C and gamma values were optimized in case of using RBF kernel but in case of linear kernel we only need to optimize value of c. Also, the effect c hyperparameter will be less when gamma is high.

Disadvantages of SVM is as follows:

- For large datasets SVM is not suitable.
- SVM will underperform if number training data records were less than the number of features for every datapoint.
- If there is more noise in dataset the performance of decision model will be affected.



In our case we used SVM to predict the dependent variable, before training the model we firstly used grid search cv to find best suitable hyperparameter values. The best hyperparameter values are c:1.0, gamma:0.2, kernel: linear and max_iter:1000.

The recorded ROC-AUC score for logistic regression in our prediction is:

H1N1 Vaccine: ***0.828***

Seasonal Vaccine: ***0.856***

Decision Tree Classifier:

Decision trees recursively split the data using a tree until one left with pure leaf nodes that is the data with only one type of class. Decision trees mainly have two types of nodes i.e., decision nodes and other is leaf nodes. The decision nodes help us to split the data and leaf nodes helps to decide class of new datapoint. The main thing in decision trees is that how we define impurities and how we classify. Let's discuss about them later but for now let's see how decision tree works and splits the dataset. Firstly, in root node it takes all the datapoints then the algorithm will be going to split the data based on the condition of the root node. Now the points which meet the condition will go to left child of the root node and which doesn't satisfy will go to the right. Here by splitting the datapoints according to the conditions will reduce the impurities of the dataset. The algorithm will be goanna to follow the same splitting rule for all the remaining nodes. After reaching the leaf nodes the main point is how to classify the dependent variable value for a given independent variables or datapoints values. Here algo will traverse the decision tree starting from the root like if the condition satisfied it goes to left and if not then right. Finally, when it reaches the leaf node, algo will assign the class which is given the datapoints in the same node. But in reality, we don't see 100% pure leaf nodes, in those cases we perform a majority voting and assign the majority class to the test point. Therefore, by following the same method one can classify the dependent variable value for any datapoint on the n-dimensions.

How do we split the data in this classification algorithm? Basically, we will be having the whole dataset and the task is to find the best splitting condition, like for example if we want to evaluate between any two conditions. Then we first split the datapoints according to those two conditions, now the question is that what is the better split? To find this we need to calculate which split is maximizing the information gain of the child nodes the most. Root node has highest impurity or uncertainty to quantify this we use entropy; it is the measure of the information contained in a state. If entropy is high the uncertainty or impurity will be high the entropy formula is given below:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

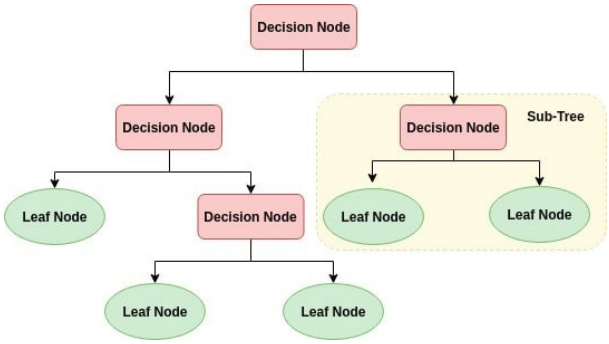
Using this formula, we calculate entropy of all the four states for those two splits. Now to find the information gain corresponding to the split we subtract the combined entropy of the child nodes from the entropy of the parent node, the weights are just the relative size of the child w.r.t parent, IG formula is given below:

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j, after)$$

By computing Information Gain (IG) for both the splits we select the split which has maximum IG value. Here we only compared for two splits but in reality, the model evaluates entropy and IG for every possible split and selects the best one.

Disadvantages of Decision Tree Classifier is as follows:

- For continuous numerical features decision tree can't be used well.
- The tree will be instable means with even the small changes in the dataset affects the tree structure.
- The training takes longer time than other regressor models.
- Another disadvantage is that the model is prone to overfit the data but it can be resolved by hyperparameter tunning.



In our case we used SVR to predict the dependent variable, before training the model we firstly used grid search cv to find best suitable hyperparameter values. The best hyperparameter values are min_samples_leaf: 9, max_features: 9, max_depth: 9, criterion: gini.

The recorded ROC-AUC score for logistic regression in our prediction is:

H1N1 Vaccine: ***0.763***

Seasonal Vaccine: ***0.829***

Random Forest Classifier:

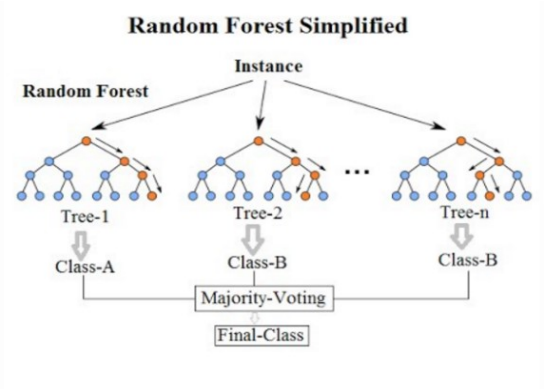
In decision trees if we change the training data slightly now for this slightly modified dataset, we will get a completely different tree. So, decisions trees are highly sensitive to the training data which could result high variance, now random forest overcomes this issue. Random forest classifier is a collection of multiple random decision trees and its less sensitive to the training data. In this model the first step is to build multiple datasets from our original data, here we are selecting rows randomly and every dataset will contain the same number of rows as the original one. For any dataset there can rows repeated that means we are performing random sampling with replacement. The process followed to create a new data is called bootstrapping. Now we train a decision tree on each of the bootstrapped datasets independently. In case of

features we won't use every feature, instead we select subset of features for each tree and use them for training. After getting data and the feature subsets we are ready to build the trees. After creating the decision trees for every set, we have to finally make a prediction/classification based on those trees. For suppose we take a new datapoint we pass the datapoint through each tree one by one and note down the predictions. As this is the classification, we consider the majority voting and classify the datapoint based on the vote results.

This process of combining results from multiple models is called as aggregation. Therefore, in random forest we perform bootstrapping and then aggregation in combined it called as bagging. Here bootstrapping helps our model to be less sensitive to the original training dataset, the random featuring selection helps to reduce the correlation between the trees, hence variance will be decreased. The ideal size of the feature subset is the value close to the log or sqrt of the total number of features. The parameters needed to be tuned in random forest are n_estimators they will be representing the frequency of trees to be used in the algorithm, adding more trees will make the model training process slower. Another parameter is max_depth which tells about depth of each tree, better information will be obtained if the depth or splits are more. Min_samples_leaf tells minimum number of samples should be there at leaf node. Max_features represent the no.of features need to be looked for best split.

Disadvantages of Random Forest Classifier:

- These doesn't work well for regression problems.
- As the number of trees increases the model becomes slower.



In our case we used RFC to predict the dependent variable, before training the model we firstly used grid search cv to find best suitable hyperparameter values. The best hyperparameter values are n_estimators: 1000, min_samples_leaf: 4, max_depth: 19.

The recorded ROC-AUC score for logistic regression in our prediction is:

H1N1 Vaccine: ***0.835***

Seasonal Vaccine: ***0.860***

Artificial Neural Networks:

Here we used multi-layer perceptron classification model as it is a multilayer, we find more than one hidden layer in the neural network. This model contains input layer connected with hidden layers and finally output layer. In the neural network input layer nodes will be fully connected to the hidden layer 1 nodes, hidden layer 1 nodes are fully connected to hidden layer 2 nodes … up to hidden layer n nodes are fully connected to output layer. Now one need to randomly initialize weights and biases to their nodes and as we all know those weights and biases are incorrect. Now our target is to make weights and biases as correct as possible so, for that first one has to do forward propagation by getting the weighted sum of inputs and then substituting that value in some activation function that they use those are like linear, sigmoid, tanh, ReLU, Leaky-ReLU etc. In this way we will do feed-forwarding up to the output layer. Now in output layer one will get some random outputs obviously those are incorrect. After that one should check the output that they got with the correct output from dataset, by this way we can get the loss after getting loss one need to do back propagation to rectify the weights and also to reduce the loss so that our neural networks work accurately. Here to change weights one must use some optimizers like Gradient Descent, Stochastic Gradient Descent, Mini- Batch Gradient Descent, Momentum, Nesterov Accelerated Gradient, Adagard, Adadelta and Adam. But actually, Adam optimizer is the best optimizer because it will train the neural network in less time and it is more efficient and in case of Gradient Descent optimizers Mini batch Gradient is best. In this way one can use optimizers and reduce the loss. One more thing while writing neural network code one need to specify the learning rate means the rate at which the model should learn so it is not a fixed value they need to configure it according to no.of hidden layer, no.of nodes in each layer, no.of epochs and the kind of activation and optimizer used. That's the basic concept of Artificial Neural Networks.

Here in case of MLP classifier we need to tune hyperparameters mainly like hidden_layer_size, max_iter, activation, solver and random_state. Here hidden_layer_size tells us the number of layers to be included in the neural network and size of each layer. To find size of hidden layers experts says to do the below shown calculations.

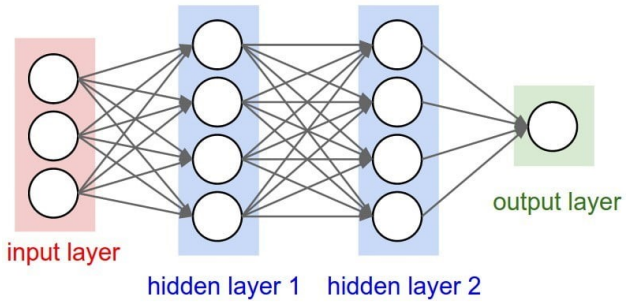
- Midway between input and output
- Less than 2x of input nodes
- 2/3*(input nodes)+output nodes

max_iter means number of epochs, solver hyperparameter tells the optimization function to be used for weight optimization.

Disadvantages of ANN:

- ANN algorithm is dependent on hardware, they require processors which does parallel processing.
- The network structure of the model is not properly determined.

- ANN is not reliable, so times we don't know on what basis does it provides solution.



In our case we used MLP classifier to predict the dependent variable, before training the model we firstly used grid search cv to find best suitable hyperparameter values. The best hyperparameter values are hidden_layer_sizes: (50,50,50), activation: logistic, solver: adam random_state: 0, max_iter: 1000.

The recorded ROC-AUC score for logistic regression in our prediction is:

H1N1 Vaccine: ***0.808***

*Seasonal Vaccine****: 0.848***

6. Description of the Dataset:

The data for this project comes from the "National 2009 H1N1 Flu Survey (NHFS)" and the public made available by Kaggle. The data set contains 3 csv files they are training_set, testing_set, training_labels.

a) The Training data Contains the: 26,706 values
b) The Testing data Contains the: 26,709 Values

Column Labels: It basically contains the 31 different columns

- **"h1n1_concern** - Level of concern about the H1N1 flu."
- **"h1n1_knowledge** - Level of knowledge about H1N1 flu."
- **"behavioral_antiviral_meds** - Has taken antiviral medications. (binary)"
- **"behavioral_avoidance** - Has avoided close contact with others with flu-like symptoms. (binary)"
- **"behavioral_face_mask** - Has bought a face mask. (binary)"
- **"behavioral_wash_hands** - Has frequently washed hands or used hand sanitizer. (binary)"
- **"behavioral_large_gatherings** - Has reduced time at large gatherings. (binary)"
- **"behavioral_outside_home** - Has reduced contact with people outside of own household. (binary)"
- **"behavioral_touch_face** - Has avoided touching eyes, nose, or mouth. (binary)"
- **"doctor_recc_h1n1** - H1N1 flu vaccine was recommended by a doctor. (binary)"
- **"doctor_recc_seasonal** - Seasonal flu vaccine was recommended by a doctor. (binary)"

- **“chronic_med_condition** - Has any of the following chronic medical conditions: asthma or another lung condition, diabetes, a heart condition, a kidney condition, sickle cell anemia or other anemia, a neurological or neuromuscular condition, a liver condition, or a weakened immune system caused by a chronic illness or by medicines taken for a chronic illness. (binary)”
- **“child_under_6_months** - Has regular close contact with a child under the age of six months. (binary)”
- **“health_worker** - Is a healthcare worker. (binary)”
- **“health_insurance** - Has health insurance. (binary)”
- **“opinion_h1n1_vacc_effective** - Respondent's opinion about H1N1 vaccine effectiveness.”
- **“opinion_h1n1_risk** - Respondent's opinion about the risk of getting sick with H1N1 flu without a vaccine.”
- **“opinion_h1n1_sick_from_vacc** - Respondent's worry of getting sick from taking the H1N1 vaccine.”
- **“opinion_seas_vacc_effective** - Respondent's opinion about seasonal flu vaccine effectiveness.”
- **“opinion_seas_risk** - Respondent's opinion about the risk of getting sick with seasonal flu without a vaccine.”
- **“opinion_seas_sick_from_vacc** - Respondent's worry of getting sick from taking the seasonal flu vaccine.”
- **“age_group** - Age group of respondents.”
- **“education** - Self-reported education level.”
- **“race** - Race of respondent.”
- **“sex** - Sex of respondent.”
- **“income_poverty** - Household annual income of respondent with respect to 2008 Census poverty thresholds.”
- **“marital_status** - Marital status of the respondent.”
- **“rent_or_own** - Housing situation of the respondent.”
- **“employment_status** - Employment status of the respondent.”
- **“hhs_geo_region** - Respondent's residence using a 10-region geographic classification defined by the U.S. Dept. of Health and Human Services. Values are represented as short random character strings.”
- **“census_msa** - Respondent's residence within metropolitan statistical areas (MSA) as defined by the U.S. Census.”
- **“household_adults** - Number of *other* adults in the household, top-coded to 3.”
- **“household_children** - Number of children in household, top-coded to 3.”
- **“employment_industry** - Type of industry respondent is employed in. Values are represented as short random character strings**.”**
- **“employment_occupation** - Type of occupation of the respondent. Values are represented as short random character string’’

7. Proposed Methodology:

i. Data Pre Processing

In this section we are going to discuss about how we designed and implemented this project in a detailed manner. Initially we did exploratory data analysis to understand the H1N1 and Seasonal flu dataset and the features dependencies and correlations etc. after that we did preprocessing to our dataset so that makes our dataset to be ready for

model building. Then after we tried and tested various machine learning classification algorithms some of them are included in Literature review section on the dataset to predict the probabilities of a person taking H1N1 and Seasonal flu vaccines.

Exploratory Data Analysis and the preprocessing:

Firstly, we imported the dataset which is of (26707, 38) size and used panda's library for doing our data analysis. In our analysis we observed Features like health_insurance, income_poverty, employment_industry and employment_occupation are having nearly 50% of null values. Also features like h1n1_knowledge and opinion_h1n1_sick_from_vacc are negatively corelated that means as the people get more knowledge about H1N1 vaccine their misassumptions on the vaccines will be reduced. Rest of the features are also having null values but their percentage is very less, so we modified them by filling with mode value of that feature, in this way we tackled null values. The above mentioned 4 features are also not much correlated with the target variables and their null percentage is near to 50 so we dropped those features from our dataset, finally dataset size is (26707, 34). As the further step of our preprocessing, we did one hot encoding to the categorical features of our dataset.

One hot encoding is a technique used to transform a categorical features data into a numerical form so that we can use it for machine learning algorithms. One hot encoding transforms every single value from a feature as one of the columns by itself. We use one hot encoding because our models don't understand the words, we use so it only can understand numerical data such as integers while training, the integer values in one hot encoding are binary in nature. This is about one hot encoding and our case we preformed it for all the categorical features or the features with object datatype. We used pandas get_dummies() method to create one hot encoding / dummy columns for a feature. After creating dummies, we dropped the columns which are categorial/object types from the dataset. Then we performed train-test split for both H1N1 and Seasonal vaccines separately with 80:20 splitting ratio. After this we performed feature scaling, it is a step done in the preprocessing step of the dataset and feature scaling normalizes the ranges of features values. Generally, every feature will be represented by magnitudes and units, here magnitudes are the values of that feature, now if we don't perform feature scaling and try to apply the same feature values on models like for suppose KNN we take, KNN uses Euclidian distance to calculate distance between datapoints. Therefore, as we have not done feature scaling the distances of the datapoints will be much more, so we have to scale down the values such that each of the feature performs some scaling techniques. We have to perform scaling techniques mandatorily to some of the machine learning algorithms which uses Euclidean distances, gradient descents etc. they are linear regression, K means clustering, KNN, deep learning algorithms like DNN, ANN, CNN and RNN. In case of ensembled algorithms like Random Forest, Decision Trees, XGBoost scaling is not mandatory. In our case we used Standard Scalar, it will scale down feature based on the standard normal distribution, it scales our feature considering mean =0 and SD=1. Formula for standard scalar is given below:

$$z = \frac{x - \mu}{\sigma}$$

Here we have done standard scaling for train and test splits for both the vaccines data. The last step of our preprocessing is that we done feature selection using Lasso regression. Lasso by decreasing the absolute values of the coefficients will optimize the cost function and prior using Lasso the features need to be scaled. The cost function of the Lasso regression is as follows:

$$\frac{1}{2N_{training}} \sum_{i=1}^{N_{training}} \left(y_{real}^{(i)} - y_{pred}^{(i)}\right)^2 + \alpha \sum_{j=1}^{n} |a_j|$$

Here alpha tunes the intensity of the l1 penalty term and a_j means the coefficient of j^{th} feature. As the coefficient of the feature value is high the value of the cost function increases. Here the main point is that as the Lasso try's to minimize the cost function, it makes some of the coefficients values as zero. Zero coefficient indicates that the feature is useless for the model training so we just have to remove or discard the features whose coefficients are set to zero. While using Lasso we first have to tune its hyperparameter that is alpha we made it to 1 in our case. After using Lasso regression, we found 6 features are useless in H1N1 vaccine prediction and 3 features are useless in Seasonal vaccine prediction. Therefore, we discarded those columns from our training datasets. In this way we have done our data analysis and preprocessing part for our dataset, the next step is model building.

ii. Training

Data test train split

A strategy for analyzing the efficiency and performance of a machine learning techniques is the train-test split. This technique can be applied for any supervised learning algorithm i.e., regression or classification. It involves the process of dataset separating it into two subgroups. The training dataset will be the first part of the split whereas the second part will be used for testing the accuracy of the trained model as it contains the data which is unknown to the model.

Training Dataset: Data set that is being used to train the ml model.

Testing Dataset: This dataset is used to assess a model's performance.

The purpose is to evaluate the machine learning effectiveness of the model on new data that hasn't been incorporated into the classifier yet. We used the train test split of 67% and 33% for better accuracy. The test train split is achieved with the help of the scikit learn package.

Loss function:

A grid termed the confusion matrix could be generated for each classification prediction models, displaying the set of test instances correctly and mistakenly categorised.

	Actual 0	Actual 1
Predicted 0	True Negatives (TN)	False Negatives (FN)
Predicted 1	False Positives (FP)	True Positives (TP)

- TN: Number of negative cases correctly classified
- TP: Number of positive cases correctly classified
- FN: Number of positive cases incorrectly classified as negative
- FP: Number of negative cases correctly classified as positive

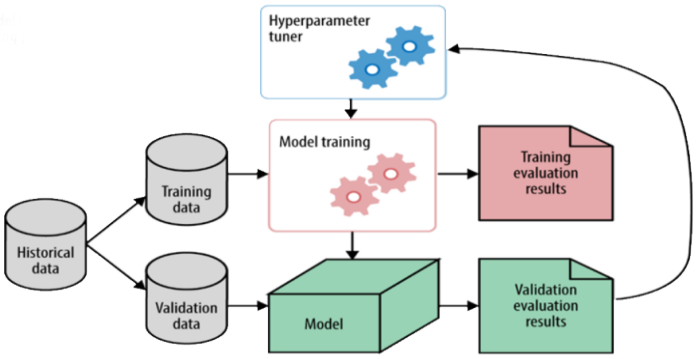
The most basic statistic is **Accuracy**, which is calculated as the amount of properly categorized test instances dividing the number of tests.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN)$$

It can be used to solve a wide range of issues, although it isn't particularly effective when dealing with imbalanced datasets. Accuracy is a false evaluation of the model's performance. We moved toward the AUC-ROC and Log loss, "ROC curve is a plot of true positive rate (recall) against false positive rate (TN / (TN+FP)). AUC-ROC stands for Area Under the Receiver Operating Characteristics and the higher the area, the better is the model performance. Log loss is a very effective classification metric and is equivalent to -1* log (likelihood function)"

Hyper parameter tuning

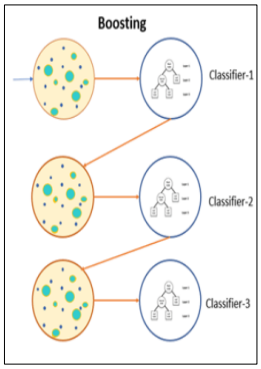
A model hyperparameter is a variable that really isn't part of the model and can't be predicted from data. On a particular situation, we can't possibly know what the optimum value for a model hyperparameter is. We can apply basic guidelines, like transfer values from other situations, or utilize guess work to get the optimal solution. We modify the model's hyperparameters to find the model's characteristics that produce the best accurate predictions. Basic techniques that can be used for hyper parameter tuning are randomized search cv and Grid search cv.

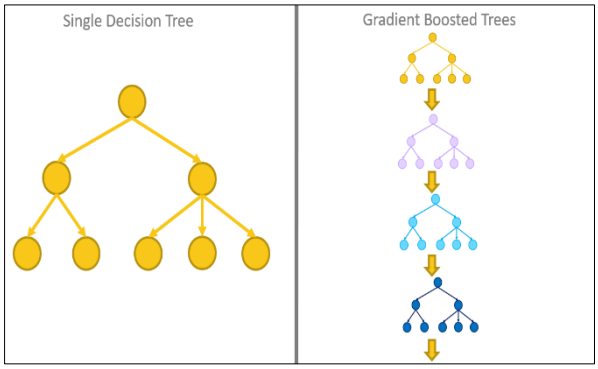


We used Optima for hyper parameter tuning. “Optuna is a software framework for automating the hyperparameter tuning process. It uses a variety of samplers to determine the best hyperparameter values, including grid search, random, Bayesian, and evolutionary methods.”

Algorithm:

We Used the Gradient Boosted Trees algorithm for predicting both the seasonal and H1N1 vaccines. To enhance accuracy, ml algorithms need much more than fitting the model & generating predictions. Ensemble Methods are now being used by the majority of successful models in the market and in contests to improve their performance. Gradient Boosting is one such approach. Boosting is a machine learning method that turns poor learners become strong ones. poor learners are classifiers that always perform somewhat better than random, regardless of the training distribution of data. The predictions in Boosting are incremental, with each following predictor learning from prior predicts' mistakes. In this area, Gradient Boosting Trees (GBT) is a widely used approach. We used GBT for this because they are Best for Heterogeneous Data (doesn't have much internal structure) Easy to Use and Works well for Small Data.





Implemented the GBT with the help of two exsisting libraries they are the XGBosst and CatBoost.

The catboost is also working on the same core principle of supervised learning i.e. minimising thed loss function L,

$$L\left(f(x), y\right) = \sum_i w_i \cdot l\left(f(x_i), y_i\right) + J(f)$$

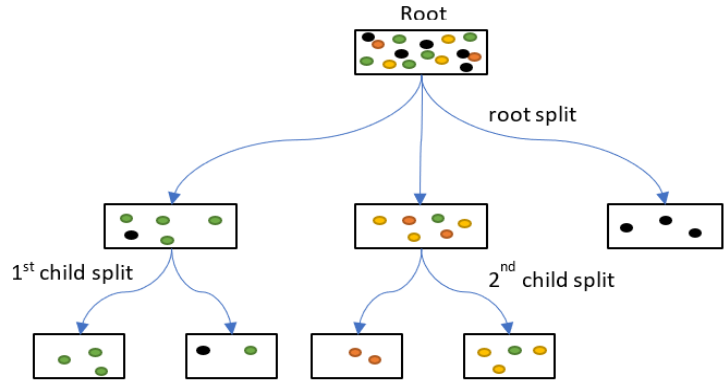
Where J(f) is the regularization term. The catboost internally uses two types of the loss function they are cosine and L2 In this case we are using the L2 technique which is used by all type of algorithms.

$$L2 = -\sum_i w_i \cdot (a_i - g_i)^2$$

Where a_i denote $f(x_i)$, w_i the weight of ith object, and g_i the corresponding gradient of l. The L2 score on the loss function will gives the following resultant equation.

$$S(a,g) = -\sum_i w_i(a_i - g_i)^2 = -\left(\sum_{i:x_{i,j}\leq c} w_i(a_{left} - g_i)^2 + \sum_{i:x_{i,j}>c} w_i(a_{right} - g_i)^2\right)$$

The optimum values of a_{left} and a_{right} will be calucualted and the the tree will be constructed to a certain depeth d repeating the same procedure



iii. Model Selection

After the models have been examined depending on the specified criteria, model selection is indeed a strategy for deciding the proper model.

Re - sampling techniques provide basic strategies for reordering samples to see whether the model is working well enough on datapoints which haven't been trained on. To put it another way, resampling allows us to see does the model is capable of generalization of data effectively.

Random Splits will be considered when we want to choose a portion of data arbitrarily and divide it into the test, train and validation data sets. The benefit of this technique would be that the original population is likely to be very well reflected in all three groupings. Random splitting, to put it another way, prevents biased data sampling.

The **K-Fold Cross-Validation** approach shuffles the data at irregular intervals and afterwards divides it into k clusters. Following that, while evaluating over each grouping, the group should be regarded a test set, while the balance of the groups should be combined into a train dataset. On these sample population the model is then tested, and the procedure is repeated for the remaining k clusters. As a result, at the conclusion of the procedure, one will have k distinct test group findings. The accurate model may then be readily chosen by selecting the model with the highest performance.

Model complexity, along with model performance, is taken into consideration using stochastic metrics. The capacity of a model to capture variation in data is measured by its complexity. A strongly biased model, such as the logistic regression technique, is much less complicated, but a neural network has a high degree of complexity. Another key issue to keep in mind would be that the performance of the model used in stochastic measurements is based only on the training set.

Based on the above-mentioned techniques we found that cat boost is working better compared to that of the other algorithms we finalized the cat boost as the best model for predicting this data.

iv. Model Preservation:

The train model is preserved in the .sav file with the help of pickle serialization in which converts any object into pickle files. These file can be used to predict the others with the help of pickle loaders.

v. web-Application:

The web application is created with the help of angular framework . The backend and API is handled by the flask and fastAPI and MongoDB is used for the storage of data. The Application allows the user to perform basic crud operations and model analysis.

Features of the web application:

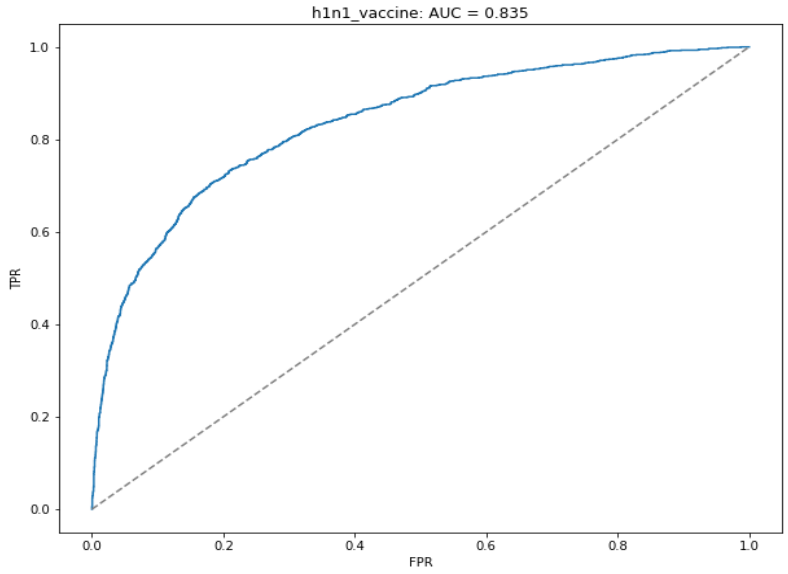
Feature	Usage
Add Project	Create a new workspace
Add Dataset	Can Upload the training data set into the server and use it for training
Build	Tarin the model, Analyze the dataset and generate reports
Predict	Can predict the vaccination probability of individual or for group of people
Deploy	Make the project available to public
CRUD Operations	Basic Crud operations on the models
Data set Analysis	Generates dataset related reports and provides the interface for performing dataset related operations

8. **Experimental Results:**

The result of cat boost model for the H1N1 vaccine is as follows:

Accuracy: `0.84`

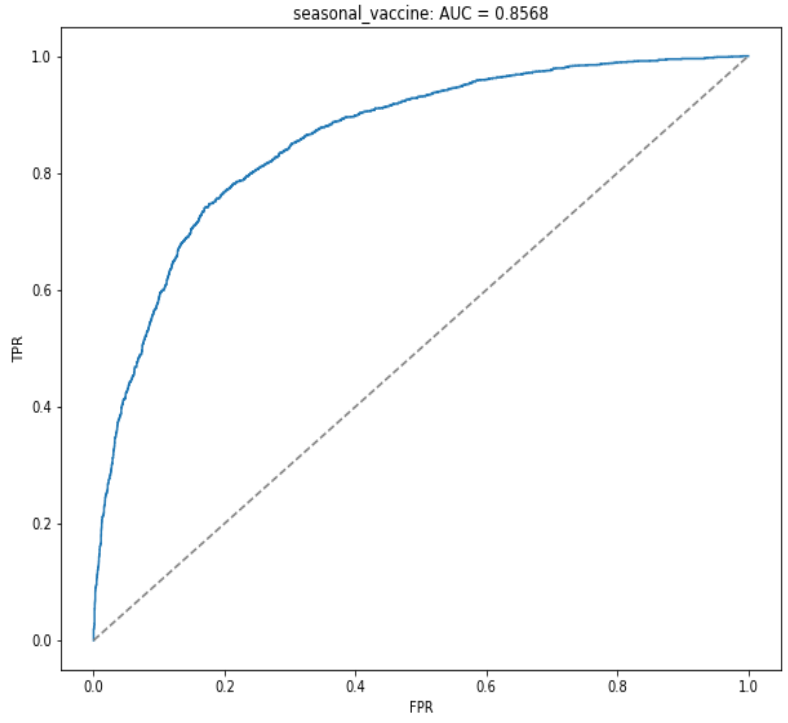
AUC score : `0.83517391078386`



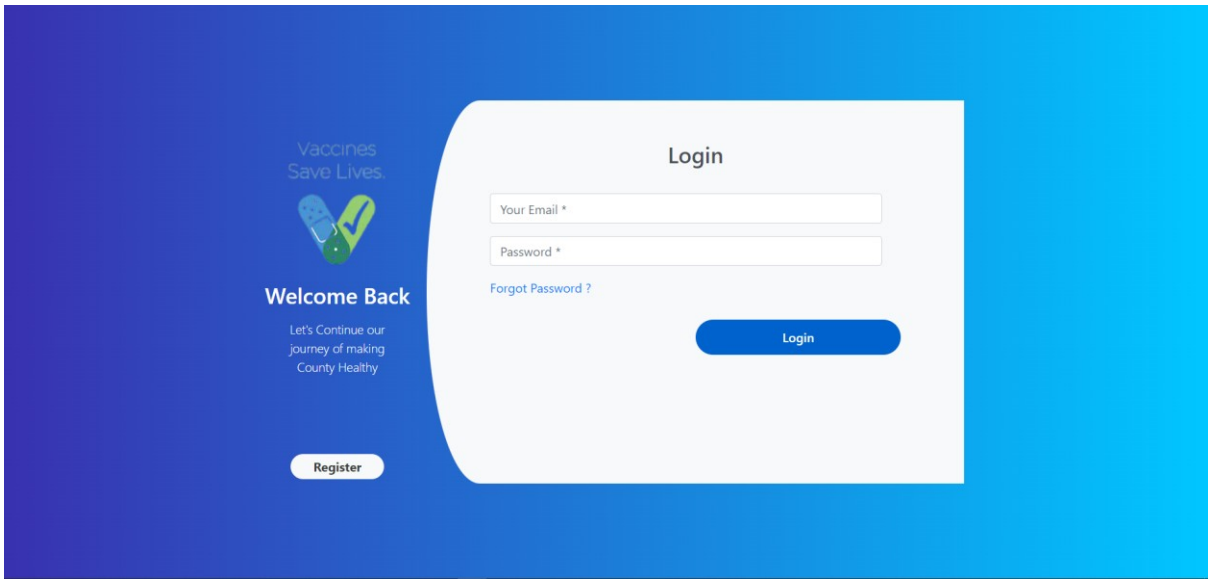
The result of cat boost model for the Seasonal vaccine is as follows:

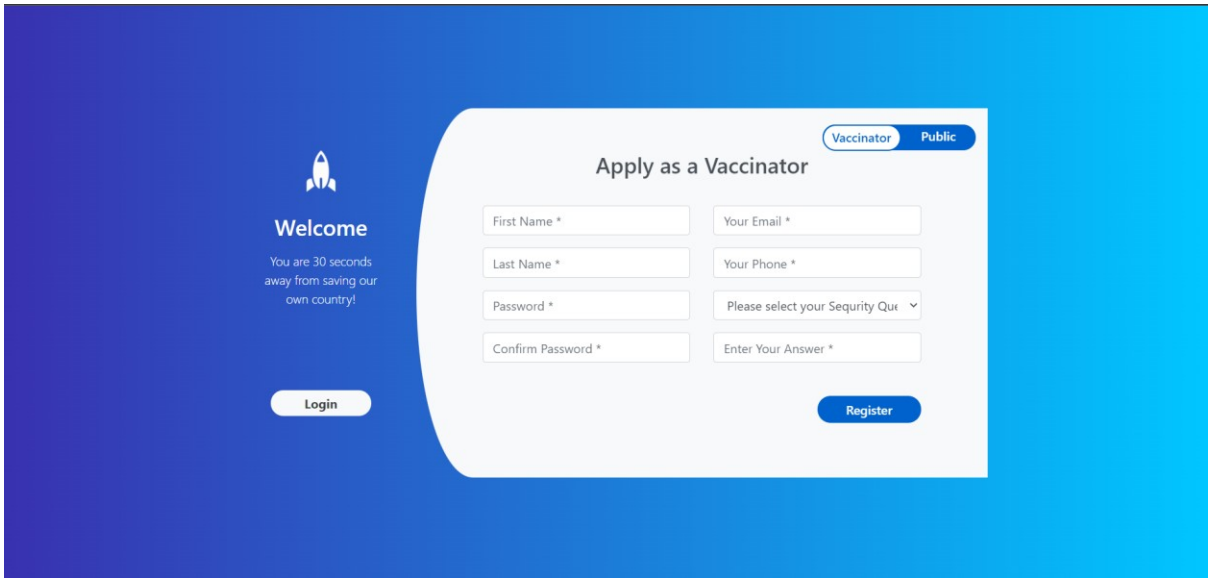
Accuracy: `0.86`

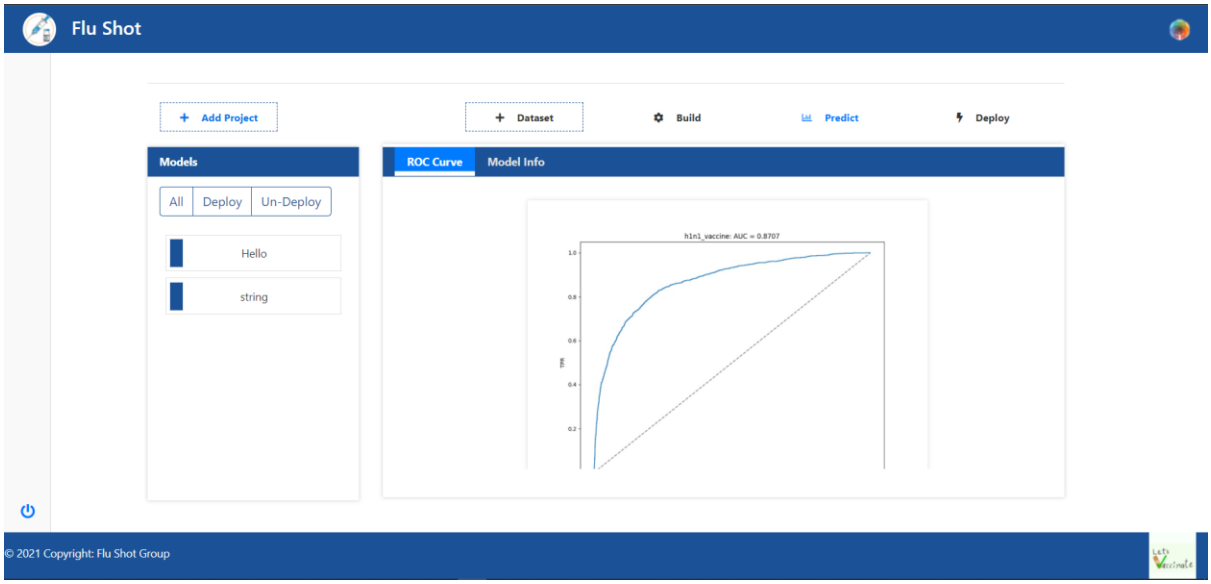
AUC score : `0.8568107485842313`

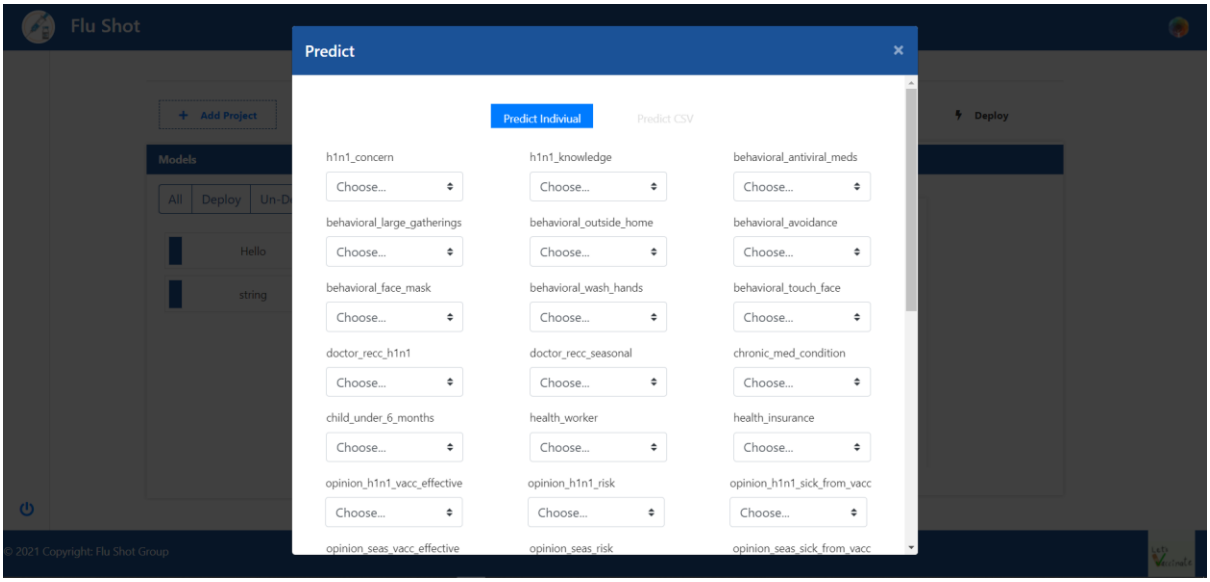


The interface of the web application:









9. Conclusions and Future Scope

Conclusion:

Vaccination are the important path of human health at least 10 vaccines are given to the people before the age of 5 years. Artificial Intelligence has proved its importance in every segment, the usage of AI in healthcare will become a major breakthrough. Our intention is to develop the platform so that it'll work for vaccination, Due to its importance in the public health we have tested different algorithms for making the platform accurate

Future Scope:

- This platform can be further developed such that it can be used to predict any type of vaccination
- By collecting the data more and more the accuracy can be further improvised.
- With the help of cloud the model can be trained repeatedly and continuously.
- Making the platform available to the public can help the vaccinators to identify the people who may not vaccinate and works as the carriers.

9. References :

- Introduction to Taylor's theorem: https://mathinsight.org/taylors_theorem_multivariable_introduction
- How to calculate gradient and hessian of log loss objective function: https://stats.stackexchange.com/questions/231220/how-to-compute-the-gradient-and-hessian-of-logarithmic-loss-question-is-based
- https://catboost.ai/en/docs/
- https://arxiv.org/pdf/1603.02754.pdf
- https://arxiv.org/abs/1810.11363v1
- https://www.kaggle.com/jeromeblanchet/flu-shot-learning-h1n1-seasonal-flu-vaccines
- https://fastapi.tiangolo.com/
- https://angular.io/docs