NNFL Mini Project

Flu Shot Learning

Anuj Manoj Shah

18104B0024

B.E. Semester VII - Electronics and Telecommunication

Subject: Neural Networks & Fuzzy Logic

Professor In-Charge: Prof. Vibha Wali

Goal

Given a data set containing the responses of thousands of people to various survey questions, for each person in the survey, to use machine learning to estimate:

1. the probability that they received the H1N1 vaccine
2. the probability that they received the Seasonal flu vaccine

Background

Vaccination is a key public health measure used to fight infectious diseases. Vaccines provide immunization for individuals, and enough immunization in a community can further reduce the spread of diseases through "herd immunity."

Beginning in spring 2009, a pandemic caused by the H1N1 influenza virus, colloquially named "swine flu," swept across the world. Researchers estimate that in the first year, it was responsible for between 151,000 to 575,000 deaths globally.

A vaccine for the H1N1 flu virus became publicly available in October 2009. In late 2009 and early 2010, the United States conducted the National 2009 H1N1 Flu Survey. This phone survey asked respondents whether they had received the H1N1 and seasonal flu vaccines, 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. A better understanding of how these characteristics are associated with personal vaccination patterns can provide guidance for future public health efforts.

Dataset

Our goal is to predict how likely individuals are to receive their H1N1 and seasonal flu vaccines. Specifically, we have to predict two probabilities: one for `h1n1_vaccine` and one for `seasonal_vaccine`. Each row in the dataset represents one person who responded to the National 2009 H1N1 Flu Survey.

Labels: For this competition, there are two target variables:

- `h1n1_vaccine` - whether the respondent received the H1N1 flu vaccine.
- `Seasonal_vaccine` - whether the respondent received the seasonal flu vaccine.

Both are binary variables: `0` = No; `1` = Yes. Some respondents didn't get either vaccine, others got only one, and some got both.

Features: We have a dataset with 36 columns. The first column `respondent_id` is a unique and random identifier. The remaining 35 features are described below:

- Binary features

    - `behavioral_antiviral_meds`. `behavioral_avoidance`. `behavioral_face_mask`. `behavioral_wash_hands`. `behavioral_large_gatherings`. `behavioral_outside_home`. `behavioral_touch_face`.
    - `doctor_recc_h1n1`. `doctor_recc_seasonal`. `chronic_med_condition`. `child_under_6_months`. `health_worker`. `health_insurance`.
    - `sex`. `marital_status`. `rent_or_own`.
- Nominal features
    - `race`. `employment_status`.
    - `hhs_geo_region`. `census_msa`. `employement_industry`. `employment_occupation`.
- Ordinal features
    - `h1n1_concern`. `h1n1_knowledge`.
    - `opinion_h1n1_vacc_effective`. `opinion_h1n1_risk`. `opinion_h1n1_sick_from_vacc`. `opinion_seas_vacc_effective`. `opinion_seas_risk`. `opinion_seas_sick_from_vacc`.
    - `age_group`. `education`. `income_poverty`.
- Ratio features
    - `household_adult`. `Household_children`.

Training the ML Model

To predict the values of `h1n1_vaccine` and one for `seasonal_vaccine`, we will use algorithms such as:

- **Random Forest**: an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees.
- **Gradient Boosting**: a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.

Program Development

Software Used

- Python: An interpreted high-level general-purpose programming language.
- Jupyter Notebooks: A web-based interactive computing platform.

Packages Used

- Pandas: A software library written for the Python programming language for data manipulation and analysis.
- Scikit-learn: A free software machine learning library for the Python programming language.

Features

- Wrote a function in Python called `data_cleaning`, which deletes columns with more than 40% null values, and in columns with less than 40% null values, it uses an imputation strategy to replace the null values with substituted values
- Wrote a function in Python called `label_encoding`, which converts non-numerical data with numerical data, so that our ML algorithms can use the data for making predictions
- Used the `XGBregressor` function from `xgboost` package to utilize the gradient boosting algorithm.
- Used the `RandomForestRegressor` function from `sklearn.ensemble` package to utilize the random forest algorithm.

Results

We used the ROC AUC curve as the metric, to evaluate the performance of our ML predictions. An AUC score of 0.5 no better than random, and an AUC score of 1.0 is a perfect model.

For the Random Forest algorithm, we got a score of:

- 0.8218

For the Gradient Boosting algorithm, we had the option to vary the hyperparameters of the model, thus multiple scores:

- 0.8290 (when max-depth = 4)
- 0.8376 (when max-depth = 4, eta = 0.01)
- 0.8578 (when max-depth = 3)

As we can see, the Gradient Boosting algorithm generally performs better than the Random Forest algorithm; and we can attain better scores by varying the hyperparameters.

References

- Driven Data (2021), Flu Shot Learning: Problem Description
  https://www.drivendata.org/competitions/66/flu-shot-learning/page/211/
- Driven Data (2021), Flu Shot: Download Dataset
  https://www.drivendata.org/competitions/66/flu-shot-learning/data/
- Wikipedia
  Random forest, Gradient boosting, Receiver operating characteristic (ROC)