BIL 470 PROJECT REPORT

Ali Azak
Senior Year Electrical and Electronical Engineering Student
TOBB ETÜ
Ankara, Turkey
aazak@etu.edu.tr

***Abstract*—This report various machine learning models will be examined in a specific problem and a full machine learning pipeline (Data preprocessing to evaluating results) will be discussed in depth.**

Keywords—machine learning, data analyzing, model evaluation

I. INTRODUCTION

Machine learning is one of the fastest growing areas in computer science. Due to its high success and applicability in practical areas, importance given this area is not foreseen to be deteriorated in next years. Computer's ability to process big data and web2 improvements in data collection made many improvements which could not be predicted. Because of these important aspects, in this report we will select a problem and use it to explore many workflows of machine learning and analyze the results.

II. PROBLEM AND GOALS

A. *Problem Description*

In this project we will basically try to predict if a person chooses to get vaccinated for h1n1 and seasonal flu based on their survey results. Link for the problem is

https://www.drivendata.org/competitions/66/flu-shot-learning/

Since the problem is a contest, we will also try to outperform other models' performance as well.

B. *Motivation*

Since all around the world struggled with COVID-19 disease last few years, we believe improving machine learning models to predict vaccination trends will be important in future pandemics and current vaccination tendencies.

C. *Goal*

When we analyze the leaderboard of the contest, we see maximum of 0.8658 AUCROC score, since for any machine learning model 0.5 AUCROC score accepted as base level. Our goal is to clearly analyze and understand the problem as well as achieving over 0.8 AUCROC score.

III. PRIOR WORKS

After doing online literature search, there can be found one conference paper and GitHub repo. It can be seen that best performance achieved with xgboost and catboost in these sources. Handling with NaN values handled via pandas fillna() function and univariate function used for feature selection.
GitHub Repo
Conference Paper

IV. DATASET FEATURES

A. *Origin of Dataset*

The data for this competition comes from the National 2009 H1N1 Flu Survey (NHFS). 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. Link for the dataset is given below.

https://www.drivendata.org/competitions/66/flu-shot-learning/data/

B. *Features of dataset*

Dataset is structured and stored in csv file. Features are not sorted and some of them can be compared to each other. Dataset consists of 35 features and 2 labels. It consists of 26707 rows which stands for every person conducted the survey. Data has 35 features and 26707 rows. These rows include NaN values, and they will be needed to process accordingly. We see that if we remove all the rows including NaN values, most of the dataset will be excluded and a lot of information will be lost. We choose other methods because of that.
As we analyze the data, it is also acknowledged that there are numerical values as well as categorical values. Since most of the machine learning models need encoding to handle categorical data, we applied necessary preprocessing methods.

V. METHODOLOGY

A. *Data Preparation*

A.1. Handling Categorical Data

Most of the machine learning models needs to transform categorical data into numerical data. As we analyze the dataset there are 12 out of 35 categorical features in dataset. To deal with them we tried 5 different encoding methods (One-hot, Binary, Mean, Frequency, Ordinal) and applied most successful two of them.

a. One Hot Encoding: In this method, for every categorical data value, separate class created consists of binary values. For example, if any categorical data can take 3 values. Instead of that column, 3 separate columns created to represent these values.
b. Binary Encoding: Binary encoding is a technique used to transform categorical data into numerical data by encoding categories as integers and then converting them into binary code.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

c. Mean Encoding: Despite the ordinal encoding's ordered nature. Mean encoding allows one to encode categorical data with random order. Data gets encoded according to its mean value of occurring.
d. Frequency Encoding: In the cases where the frequency is related somewhat with the target variable, it helps the model to understand and assign the weight in direct and inverse proportion, depending on the nature of the data.
e. Ordinal Encoding: Encodes categorical features as an integer array.

After trying these various methods, state of the art dirty_cat encoding library used. Library helps to handle dirty data such as spelling mistakes, abbrevations etc. It helps using different encoding methods on appropriate data. For example, with low-cardinality data uses one-hot encoding and otherwise frequency encoding.

A.2. Handling NaN Values

In order to handle NaN values sklearn Simple Imputer, Iterative Imputer and KNN Imputer tried.

a. Simple Imputer: Missing values filled with mean values of those features. mean,
b. Iterative Imputer: Models each feature with missing values as a function of other features and uses that estimate for imputation. It does so in an iterated round-robin fashion: at each step, a feature column is designated as output y and the other feature columns are treated as inputs X. A regressor is fit on (X, y) for known y. Then, the regressor is used to predict the missing values of y. This is done for each feature in an iterative fashion
c. KNN Imputer: Each sample's missing values are imputed using the mean value from n_neighbors nearest neighbors found in the training set. Different neighbor's parameter tried but best result achieved was same with simple imputer. According to Ockham's razor, choice was made.

Dataset consists of random people conveying a survey. Due to this independency of data, KNN Imputer and Iterative Imputer did not perform well because they use other people's survey results as an input to fill missing data. Best result achieved with Simple Imputer.

A.3. Feature Selection

Selective Feature Selection method used to decide which features to use throughout the project. Figure 2 shows that using all features gives the best result.

A.4. Splitting Dataset

2/3 of train data used for training and 1/3 of train data used for testing. K cross validation used in grid search algorithm.

VI. MODELLING

Training the data realized using 6 different machine learning models. Logistic Regression, Xgboost, Catboost, Random forest, Support Vector Machines, Multi-Layer Perceptron. All these models explained briefly below, and results compared.

a. Logistic Regression:

In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure

b. Xgboost:

Gradient boosting gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. Xgboost used throughout this project to achieve a clean, fast and efficient model implementation.

c. Catboost:

Gradient boosting method same as Xgboost. Difference is in Catboost, decision trees are symmetric and splitting method is greedy despite of XGBoost histogram-based algorithm.

d. Random forest:

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.

e. Support Vector Machines:

Support vector machines use hyperplanes with lower dimension than dataset dimension. Hyperplanes found using support vectors, which are basically, data points with higher margin. These found hyperplanes are used to predict the data.

f. Multi-Layer Perceptron:

Multi-Layer Perceptron is a fully connected class of feedforward artificial neural network. Neural network consists of layers of neurons which are linear functions with learnable parameters.

VII. YPERPARAMETER TUNING

Hyperparameter tuning is one of the most important parts of a MLOps. With Hyperparameter tuning, best potential of each model may be realized and used. During the project Grid Search algorithm used for determining best hyperparameters. Models and tuned hyperparameters listed below.

A. Logistic Regression

a) Penalty: Different parameters tried between l1, l2 and elasticnet.

b) C: Each of the values in C describes the inverse of regularization strength.

c) Solver: Decides which algorithm to use in optimization.

B. Xgboost

a) max depth : Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.

b) learning rate : Step size shrinkage used in update to prevent overfitting. Higher learning rate may be lead to long converge time and lower learning rate may be lead to local minima point.

c) n estimators : It defines the number of decision trees in boosting model.

C. Catboost

a) l2_leaf_reg:Coefficient at the L2 regularization term of the cost function

b) bagging_temperature: Defines the settings of the Bayesian bootstrap. The higher the value the more aggressive the bagging is.

D. Random Forest

a) n_estimators:Number of tree in forest.

b) max_features:Number of features to look for stooping condition. If sqrt sqrt(n_features) used for example.

c) max_depth:Max depth of trees.

d) criterion: The function to measure the quality of a split. E.g. gini,entropy etc.

E. Support Vector Machines

*a) Gamma: Kernel coefficient of SVM. If 'auto', gamma = 1/n_features; if 'scale', gamma = 1/(n_features *X_var())*

b) Kernel: Specifies the kernel type to be used in the algorithm.

F. Multi Layer Perceptron

a) Hidden Layer Size: Number of hidden layers.

b) Activation Function

c) Learning_rate

When scoring the models, stratified k-fold cross validation method with k=5 and used multi output classifier to evaluate both labels together. Multioutput classifiers take average of labels' auc score and gives those values as final score.

VIII. RESULTS

Sklearn grid search algorithm used for finding the best parameters. Best results achieved in Xgboost algorithm and extra number of parameter tuning realized for better results in competition.

A. Logistic Regression

Best results achieved with below parameters.

{'C': 0.1, 'penalty': 'l1', 'solver': 'saga'}

Obtained AUC score is 0.8453315892387309.

B. Xgboost

Best results achieved with below parameters.

{'learning_rate': 0.05, 'max_depth': 5, 'n_estimators': 180}

Obtained AUC score is 0. 866989399054408

C. Catboost

Best results achieved with below parameters.
{'bagging_temperature': 0, 'l2_leaf_reg': 5}

Obtained AUC score is 0.8653315569588726

D. Random Forest Classifier

Best results achieved with below parameters.
'criterion': 'entropy',
'max_depth': 8,
'max_features': 'sqrt',
'n_estimators': 500

ObtainedAUC score is 0.8523359096027563

E. Support Vector Machine

Best results achieved with below parameters.

{'gamma': 'scale', 'kernel': 'linear'}

Obtained AUC score is 0.8379142495369231.

F. Multi Layer Perceptron

Best results achieved with below parameters.

{'hidden_layer_size': 200, 'activation_function': 'adam', 'learning_rate': 0.001 }

Obtained AUC score is 0.8235466846535454564.

IX. CONCLUSION

Different data preprocessing techniques tried. Nan values filled with mean simple imputation and one-hot encoding gave the best results. Boosting methods and random forest gave the best results. It can be said that tree-based algorithms performed well on task. After enormous of effort giving to find best hyperparameters, XGBoost gave best AUC score. Success of XGBoost can be explained by various reasons. It has effective tree pruning and overfit prevention mechanisms. Implemented in cpp to provide fast algorithm. Uses regularization techniques to achieve the best performance. In conclusion, XGBoost is usually best choice for supervised learning of well-prepared data, thanks to advantages of boosting algorithms. Deep learning methods did not try in this work, due to the complexity. Future work may explore deep learning methods performances on the dataset.

With 0.8626 score, 127th rank in 4207 competitors obtained at contest in 25.07.2022

X. APPENDIX - VISUALS

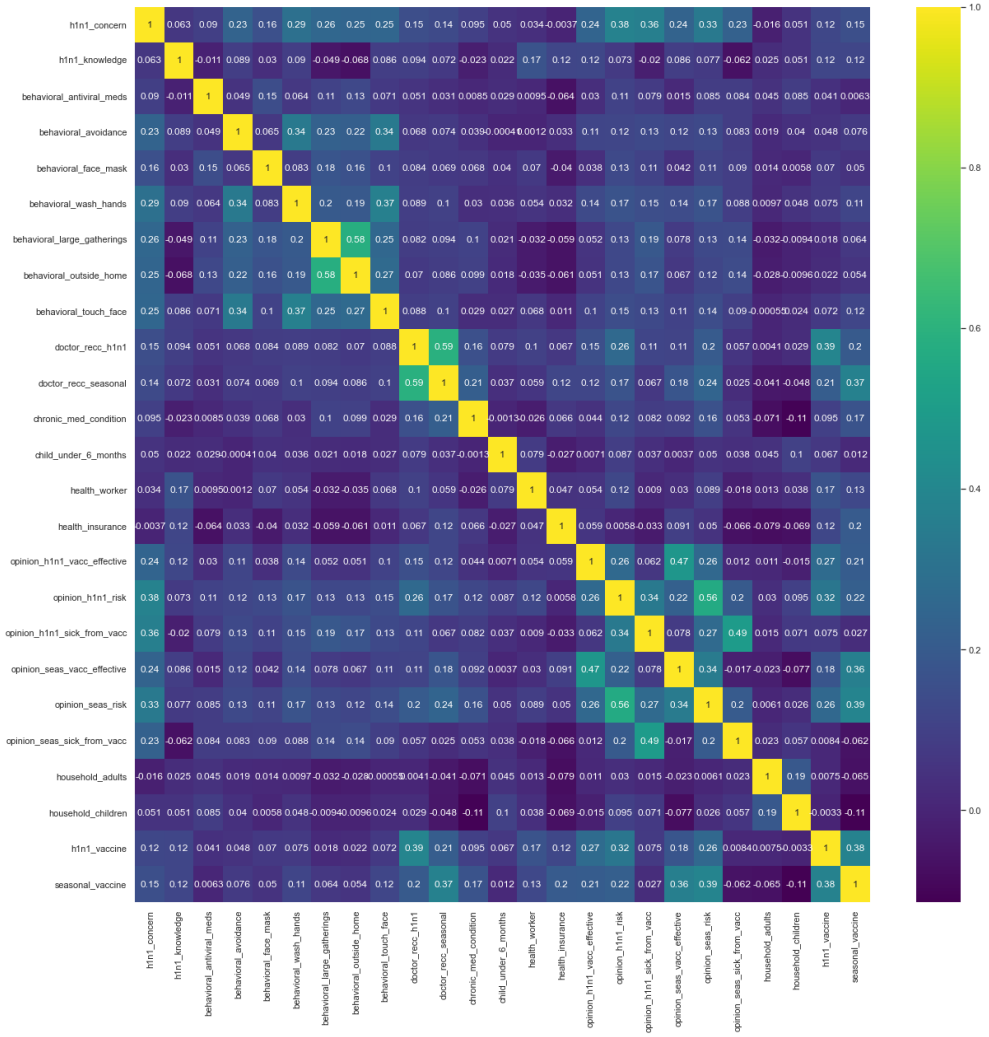

Figure 1 Correlation Matrix of Dataset

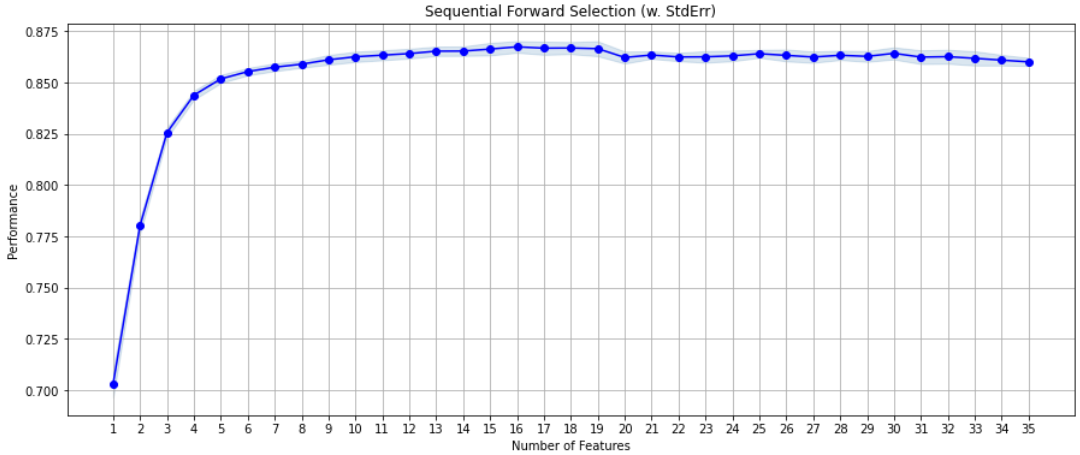

Figure 2 Feature Selection Performance

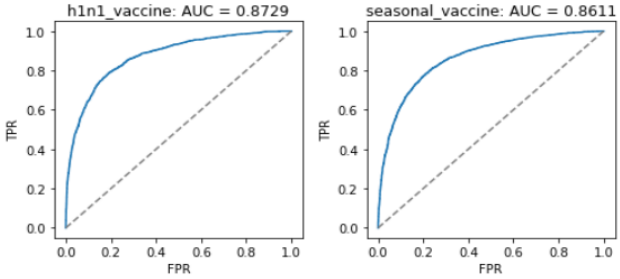

Figure 3 ROC for XGBoost

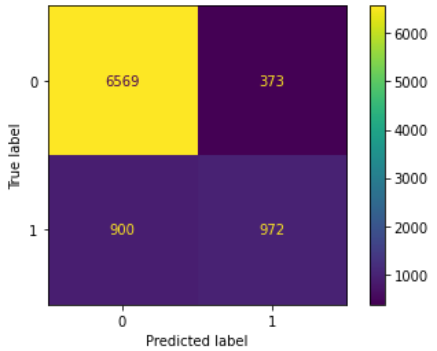

Figure 4 Confusion matrix for XGBoost h1n1 vaccine

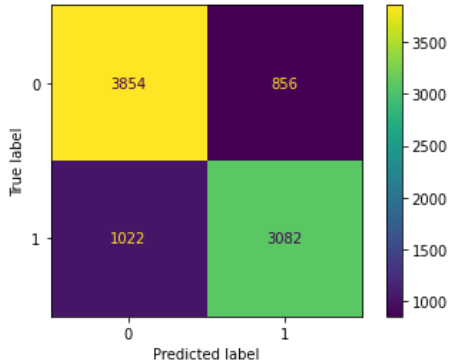

Figure 5 Confusion matrix for XGBoost seasonal vaccine

XI. References

https://en.wikipedia.org/wiki/Gradient_boosting
https://catboost.ai/
https://en.wikipedia.org/wiki/Random_forest
https://scikitlearn.org/stable/modules/preprocessing.html#preprocessing
https://scikitlearn.org/stable/modules/impute.html#impute
https://scikitlearn.org/stable/modules/linear_model.html#linear-model
https://scikitlearn.org/stable/modules/model_evaluation.html#roc-metrics

Github Repo of project:
https://github.com/aliazak6/Flu-Shot-Learning