Original Research Paper

Volume : 5 | Issue : 9 | September 2016 • ISSN No 2277 - 8179 | IF : 3.508 | IC Value : 69.48

Predicting Patient Readmissions within 30 days

Engineering

KEYWORDS :

Niranjan Singh — Research scholar, Dayanand Sagar Engineering(VTU)

Y.S Kumaraswamy — PHD (Nagarjuna Engineering College)

ABSTRACT

Readmissions of patients within 30 days of discharge cost millions of dollars of tax payers' money and pose a significant challenge to the quality of healthcare service providers.

In order to mitigate the problem posed by unwanted readmissions of patients for the same diagnosis and to increase the quality of healthcare services we seek to develop machine learning models to predict patients who are likely to get readmitted within 30 days of discharge. This research paper describes the methodologies used to build the predictive models for predicting the likelihood of patients getting readmitted within 30 days of discharge.

The work presented here provides the process to build various predictive models for predicting the likelihood of patients getting readmitted and in the process compare the accuracies of the various predictive models side by side and provide ways to improve the accuracies of the various predictive models .This paper further seeks to provide insights into the variables or the predictors which contribute the most in predicting the target variable.

1. Introduction

The problem of readmissions of patients within 30 days of discharge is not only a costly proposition but also is directly linked to the inefficiencies of the treatment process. Negligence while treating the patients by the healthcare service providers could prove fatal for the patients. Patients deserve better treatment by the service providers . The Medicare Payment Advisory Commission in the US has in the past years have laid out certain rules and regulations for the health service providers to comply with the rules and regulations failing which penalties will be imposed.

Healthcare service providers have now come up with novel ways to predict in advance the patients who are most likely to get readmitted before the readmissions actually take place. A few years ago health service providers where adopting manual methods to predict which patients are likely to get readmitted. But now more and more service providers are now adopting machine learning approaches to help predict the likelihood of patients getting readmitted.

This paper outlines a step by step process on how to build various predictive models to predict the likelihood of patients getting readmitted using machine learning process.

To help predict the high risk patients who might get readmitted we develop predictive models using Scikit-learn, a Python machine learning framework along with Apache Hive and Pandas, both open source tools for processing the files.

We then compare the performance of different learning algorithms using confusion matrix and AUC score and choose the models having high accuracies. This paper also captures the features or the variables which are most likely to help in the prediction process. Also we conduct how we might increase the accuracy by adopting various techniques. We study the characteristics of the various predictors of the patient's data such as the demographics , visit types and lab test to name a few and see how these predictors play a role in the accuracy of the prediction .

2.Previous Work

Previous work on predicting readmissions within 30 days have used different models to predict the outcome.

The approaches by Sharath Cholleti, PhD,[1] Andrew Post, MD, PhD,[1] Jingjing Gao, PhD,[1] Xia Lin, PhD,[1] William Bornstein, MD, PhD,[2] Dedra Cantrell, RN,[3] and Joel Saltz, MD, PhD[1] have provided a way that Random Forest are a good model to use in the prediction but has not explored different ensemble models in their approach. In this approach we have taken various inputs but have explored the ensemble methods by different learning algorithms to be used in the model prediction and have gone ahead to show that using ensemble methods of other learning algorithms like the SVM, logistics regression and using a weighted model helped in increasing the learning score of the models.

3.Data Manipulation

Historical data for two years were provided to build the model for future prediction of patients likely to get readmitted. As the data for the patients were contained in various files features which are likely to help in the prediction were extracted from various files and merged using existing machine learning framework tools.

Before the models could be built a lot of pre-processing has to be conducted before the data could be fed into the machine learning algorithms . After studying the various files containing the Vitals data, Lab results data, patient demographics data , the features which are most likely to contribute to the prediction accuracy of the model were carefully chosen and those data were carefully cleaned and processed.

The Data processing stage include discovering of statistical central tendencies of the various variables , the missing values had to be imputed and find out the correlation of variables both univariate and bivariate data. Once the data was properly cleaned and processed Hive query was then used to label the target value or the outcome value for the training data. The logic of whether a patient was readmitted or not consist of finding whether a particular patient was admitted for at least 24 hours and then discharged and then the same patient visits again within thirty days of his discharge only then it would be considered as readmitted. Once the labelling of data then a total of five files were generated from these pre-processing step.

```
import pandas as pd

visits = pd.read_csv('../input/prev_visit_types.csv')
```

```
vitals = pd.read_csv('../input/vitals.csv')

demo = pd.read_csv('../input/demographics.csv')

los = pd.read_csv('../input/length_of_stay.csv')

lab = pd.read_csv('../input/labdata.csv')

hos = pd.read_csv('../input/hospitalization.csv')

diag_count = pd.read_csv('../input/diagnosis_count.csv')
```

The pre-processing step required a lot of joins to be performed to get the columns details of a particular patient. Below a snippet of code is given in Python on how to perform joins, where df is the resulting data frame, pd is the Pandas reference , hos and demo are data frames for hospitalizations and demo as demographic data. The join is done on patient id and left identifies a left outer join.

```
df= pd.merge(hos, demo, on='patientId', how='left')
```

4.Feature Engineering

The previous visit types contain how many times the patient had visited earlier and for what purpose he visited ,which include emergency, outpatient or inpatient . The demographic information contain data that describes the age, sex of the patient. The lab test results include the medical treatment details. The length of stay is applicable only to those patients who got admitted before. The medical history report of each individual patient data contain details whether the patient has the history of being a smoker, non-smoker , alcoholic or whether the patient was mentally ill or not. These are the vital information that need to be captured while predicting the likelihood of the patient getting readmitted. file contains record of each patient visit history such as whether a patient who has visited earlier consisted of how many inpatient, outpatient or emergency.

With the features being extracted the next step was to do feature engineering to transform the values into numerical values so that the machine learning algorithms can work .The vitals file contained Vital Sign as well as Normal or abnormal as textual value. This was encoded as Boolean value as Normal was assigned the value 1 ad abnormal value was assigned the value 0. Features such as high risk, low risk and average risk after analysing depending on whether a patient was a heavy smoker or occasionally smoker and based on the vitals report was also encoded into numerical values. The previous visit types such as inpatient, outpatient and emergency was also given a value depending on the type of previous visit.

The type of diagnosis was also given a score based on high risk type of diagnosis. Some of the diagnosis consist of 'PH YSICALTHERAPY','PNEUMONIA','MALAISE / FATIGUE ','HYPONATREMIA','CEREBRAL INFARCT','CHRONIC KIDNEY DISEASE','HEART DISEASE'.

5.Model Building and Evaluation

Once the data was process and cleaned and then feature engineered multiple models were built on the data sets using various parameter settings. Gradient Boosting models and Random Forest models were the most powerful individual algorithm as compared to other learning algorithms. The basic outline for building the models for various learning algorithms are given below . The code below are used as a template for various other learning models.

```
fromsklearn import ensemble

fromsklearn.cross_validation import train_test_split

df = pd.read_csv('../input/readmissions.csv', sep=',')

y = df[target]

X = df.drop(target, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y)

clf = ensemble.RandomForestClassifier(n_estimators=00)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

Six different predictive models are developed using the train test split available in scikit-learn. Gradient Boosting, Support Vector Machine, Random Forest, Logistic Regression-Nearest Neighbour and Ridge Classifier models were created. The performance of each model is calculated using the confusion matrix.

Figure 1

The precision and recall is calculated for all the classifiers and then F1 score calculated. Gradient Boosting classifier and Random Forest Classifier have given a better score relative to other models after running multiple times on different datasets. The feature importance graph is shown below. Length of stay turns out to be one feature which is contributing the most in the prediction relative to the other variables.

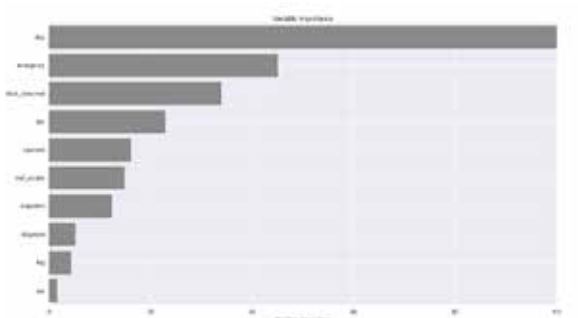

Figure 2

6.Results and Discussion

In generally almost all the models have not given extraordinary accuracy results, but Random Forest Classifier was chosen as the best model after multiple tests. An AUC score of 0.73 given by Random Forest model.

Volume : 5 | Issue : 9 | September 2016 • ISSN No 2277 - 8179 | IF : 3.508 | IC Value : 69.48

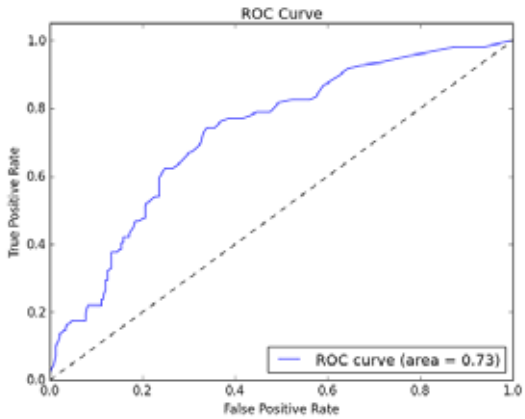

Figure 3

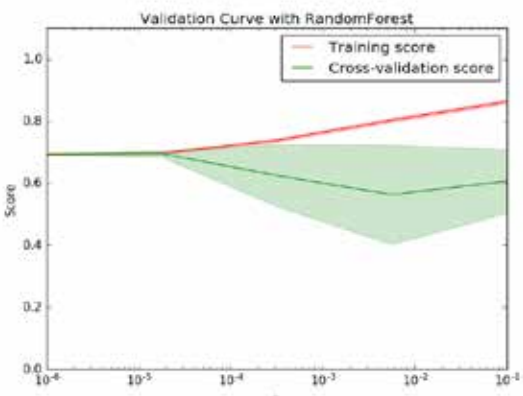

Figure 4

The figure above shows the training and the validation score curve. As there is a huge gap between the training and the validation curve the Random Forest model was overfitting as there was a lot of variance when the model was applied to various datasets.

Standard steps such as regularization, increasing the number of samples were applied to reduce the overfitting problem. But the learning curve which is given below suggest that increase in the number of samples was not going to completely solve the overfitting problem.

Figure 5

The results from the learning curve show for Random Forest has a bit of both high bias and high variance .

As seen in the learning curve above increase in the number of samples would not have either increased the accuracy of the prediction and solve the overfitting problem as described in the above paragraph . The learning curve suggest that a proper featuring engineering approach would rather help the accuracy rather than increasing the number of sample size .

7.Performance improvement using ensemble technique

Ensemble methods are techniques that combine multiple models to produce single model in order to improve the performance of the prediction accuracy . Ensemble methods usually produce more accurate solutions than a single model would produce. There are various ways to build the ensemble models but in this paper we use the weighted voting method to improve the performance by combining the weak individual models by applying a weight and then combining the results to build a single model by including the strengths of each weak learner.

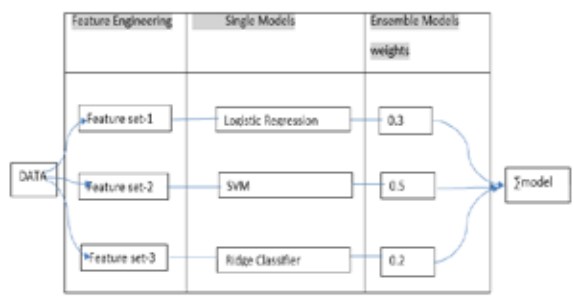

Figure 6

Every model makes a prediction (votes) for each test instance and the final output prediction is the summation of the weighted model of the individual models.

Here we show how in pseudo code how to use weighted ensemble method by using a majority vote. This has resulted in a slight increase in the AUC score and thus the accuracy of the model.

```
train = load_csv("train.csv")

target = train["target"]

test = load_csv("test.csv")

weight =[w1,w2,w3]

classifier = [svm,random forest, decision_tree_classification, ...]

predictions = matrix(row_length=len(target), column_length=len(algorithms))

for i,algorithm in enumerate(algorithms):

predictions[,i] = algorithm.fit(train, target).predict(test)*weigth[i]
```

The above pseudo code shows the algorithm of using ensemble models to achieve higher accuracy than the one we got from a single model. The technique is outlined in [9] and thus we were able to achieve a slightly better result than the one we achieved from the single model.

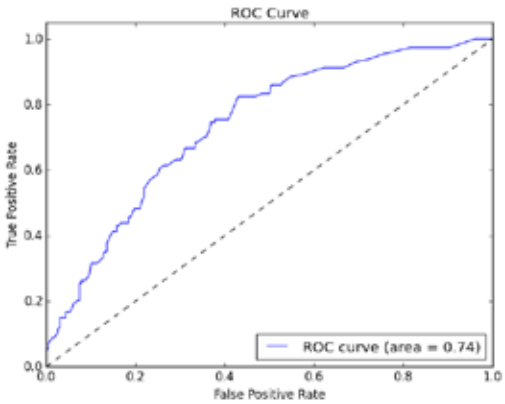

Figure 7

After applying the ensemble model the AUC score increased by not much but increased to 0.74 from 0.73, which is a little improvement although not much.

8.Conclusion and Future Work

This paper has shown that individual models when combined together to form a single model using ensemble methods perform better in terms of accuracy. In this paper weighted ensemble models was used , but there are other ensemble techniques which could have been used such as the stacked generalization method . Almost all the ensemble techniques use a common feature and that is to combine weak learners and build a single model combining the weak learners. This sort of ensemble technique was used in the most famous Netflix competition[26].

We hope that this paper has outlined the various steps from the data exploration steps to the model building step and then to the evaluation of the models and finally combining the weak learners to build and ensemble model to better the performance of individual model.

Stacked generalization ensemble technique which is another form of ensemble model could have been used and its performance could have been compared to other ensemble techniques.

References

[1] Kaggle-https://inclass.kaggle.com/c/predicting-30-day-hospital-readmissions/data

[2] National Institutes of Health-http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540449/

[3] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.

[4] Andy Liaw and Matthew Wiener. Classification and regression by Random forest. R News, 2(3):18–22, 2002.

[5] LeoraHorwitz et al. Hospital-wide all-cause unplanned readmission measure. final technical report. Centers for Medicare and Medicaid Services., 2012.

[6] Amarasingham R, Moore BJ, Tabak YP, Drazner MH, Clark CA, Zhang S, et al. An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. Medical care. 2010;48(11):981–8. Epub 2010/10/14. [PubMed]

[7] Kim H, Ross JS, Melkus GD, Zhao Z, Boockvar K. Scheduled and unscheduled hospital readmissions among patients with diabetes. The American journal of managed care. 2010;16(10):760–7. Epub 2010/10/23. [PMC free article][PubMed]

[8] Keenan PS, Normand SL, Lin Z, Drye EE, Bhat KR, Ross JS, et al. An administrative claims measure suitable for profiling hospital performance on the basis of 30-day all-cause readmission rates among patients with heart failure. Circulation Cardiovascular quality and outcomes. 2008;1(1):29–37. Epub 2008/09/01. [PubMed]

[9] http://mlwave.com/kaggle-ensembling-guide/

[10] Reducing hospital readmissions. By Jenny Minott

[11] Catlin, A. et al. "National Health Spending in 2006: A Year of Change for Prescription Drugs," Health Affairs, January/February 2008, Vol.27, No. 1, pp. 14-29.

[12] Medicare Payment Advisory Commission. 2007. Report to the Congress: Promoting Greater Efficiency in Medicare. Washington, DC: Medicare Payment Advisory Commission, p. 103.

[13] Can readmission rates be used as an outcome indicator? Ruairidh Milne, Aileen Clarke

[14] Analysis of a Random Forests Model by Gerard Biau

[15] Random Forests and Decision Trees by Jehad Ali .Rehanullah Khan, Nasir Ahmad and Imran Maqsood

[16] Classification and Regression by randomForest AndyLiaw and Matthew Wiener

[17] Learning to Rank Using Classification and Gradient Boosting by Ping Li, Christopher J.C.Burges and Qiang Wu

[18] Gradient Boosted Feature Selection by Zhixiang (Eddie) Xu , Gao Huang et. al

[19] Ensemble Classifiers and Their Applications: A Review by Akhlaqur Rahman and Sumaira Tasnim

[20] Ensemble-based classifiers by Lior Rokach

[21] Ensemble Classifiers for Steganalysis of Digital Media by Jan Kodovský, Jessica Fridrich and Vojtěch Holub

[22] Quinlan, J. R. C4.5: Programs for Machine Learning ,Morgan Kaufmann Publishers, 1993.

[23] http://en.wikipedia.org/wiki/C4.5_algorithm

[24] Report from Pike research, http://www.pikeresearch.com/research/smart-grid- dataanalytics

[25] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#prox Symposium, volume 1, July, 2005.

[26] Ensemble Learning Better Predictions Through Diversity by Todd Holloway ETech 2008