Team 10
1. Anzal Zia Khan
2. Anowarul Kabir

Introduction

During an encounter of a patient, whether his/her medications will be changed or not and he/she will be readmitted in the hospital or not, we will try to answer these questions in this project. These patients are encountered as having same problem namely ‘diabetes’. To do this, we will pre-process and clean the dataset (e.g. clean the missing values and so on) containing all the information during an encounter. Then we will select important attributes to run data mining techniques for prediction such that a future encounter of a patient can possibly know the percentage of his/her readmission in the hospital or change in the medications.

Data description

Name: Diabetes 130-US hospitals for years 1999-2008 Data Set
Source: UCI Machine Learning Repository[5]
Dataset information: The dataset has ten years of clinical outcomes at 130 US hospitals of 10000 patients. It includes 56 attributes with two class attributes namely “medication changed” and “readmitted”. All data are collected during inpatient encounter. The length of the patient’s stay at he hospital was at least one day. Additionally, the results of the lab tests during the encounter are included in the dataset.

Data preparation

1. The data contains missing values. We will clean (and fill up some) missing values so that it can not harm the model to learn the pattern.
2. No annotation required as the data is already annotated.
3. No normalization required as data is both numerical and categorical.

Game of features

We call this as game of features because this involves feature extraction, creation and selection. In this context, we will use the following techniques:
1. Histogram to understand the distribution of the attribute-values.
2. Scatter plot to find feature-feature interaction over class attributes.
3. Principal Component Analysis (PCA) for finding the maximum variant features that contributes more on class attributes.
4. Correlations among attributes.
To do this we will “scikit-learn”, a machine learning package of Python.

Methodology

We intend to use the following methodologies to solve our problem:
1) Decision Tree: A **decision tree** is a supervised learning algorithm used in classification problems. It works for both continuous and categorical data. We split

the population into two or more homogenous sets based on the most significant splitter attribute.

Decision Tree Algorithms for Splitting:

a) **Gini Index:** Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure. It works only for binary splits.

b) **Information Gain:** Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% — 50%), it has entropy of one. Entropy is also used with categorical target variable. It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

2) Support Vector Machine: A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

3) Naïve Bayes Algorithm: Naive Bayes algorithm can be defined as a supervised classification algorithm which is based on **Bayes theorem** with an assumption of **independence among features.** It is easier to understand and can be trained on a small dataset.

Metrics and performance evaluation

We intend to use the following metrics for performance evaluation:

a) Classification Accuracy: It is the number of correct predictions to the number of input samples.

Accuracy = Number of Correct Predictions / Total Number of Predictions Made

b) Logarithmic Loss: Logarithmic Loss or Log Loss, works by penalising the false classifications. It works well for multi-class classification. When working with Log Loss, the classifier must assign probability to each class for all the samples. Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below:

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

where,

y_ij, indicates whether sample i belongs to class j or not

p_ij, indicates the probability of sample i belonging to class j

c) F1 Score: F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 Score tries to find the balance between precision and recall.

- **Precision :** It is the number of correct positive results divided by the number of positive results predicted by the classifier.
  Precision = TruePositives / (TruePositives + FalsePositives)

- **Recall :** It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).
  Recall = TruePositives / (TruePositives + FalseNegatives)

Discussions and conclusions

We intend to train our model using the algorithms discussed in this report and calculate its performance against the test sample using the performance metrics discusses in this report.

Milestones

a) Data Preprocessing – 20th October
b) Training and Testing the Model – 14th November
c) Presentation and Report – 28th November

[1] https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[2] https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb
[3] https://medium.com/@srishtisawla/introduction-to-naive-bayes-for-classification-baefefb43a2d
[4] https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72
[5] https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008