IJDCST @ May-June-2021, Issue- V-7, I-1, SW-20
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

PREDICTION OF HEART ATTACK USING SPARK AND DEEP LEARNING

CH. PRANEETH* D. MADHUSUDANRAO**
***ASSISTANT PROFESSOR, IT DEPT, PVP SIDDHARTHA INSTITUTE OF TECHNOLOGY**
**** ASSISTANT PROFESSOR, IT DEPT, VFSTR**

ABSTRACT

All these can be taken into consideration and even more reliable and more accurate algorithms can be used. We can make use of more learning techniques and one of the big data tools which is spark to predict heart attack chances with less time and more accuracy. Then the project will be more powerful to depend upon and even more efficient to depend upon.By using different kinds of machine learning and deep learning techniques to predict the possibility of occurrence of heart disease have summarized. Determined the prediction accuracy of each algorithm and apply the proposed system for the area it needed.

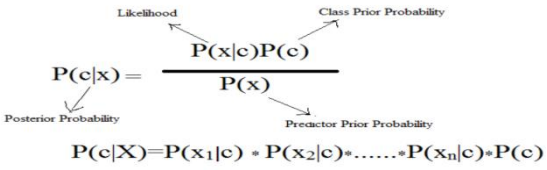
INTRODUCTION

It is a classification technique based on Bayes theorem with an presumption of independence among predictors. In basic terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For illustration, a fruit may be considered to be an apple if it is red, circular, and almost 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple which is why it is known as 'Naive'.

Naive Bayes model is simple to build and particularly useful for huge data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem gives a way of calculating posterior probability P(c/x) from

P(c), P(x) and P(x/c). Look at the equation below:



Above,

- *P*(*c/x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class*.

IJDCST @ May-June-2021, Issue- V-7, I-1, SW-20
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

- *P*(*x/c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

HOW NAIVE BAYES ALGORITHM WORKS?

Let's understand it using an example. Below I have a training data set of climate and corresponding target variable 'Play' (recommending conceivable outcomes of playing). Now, we need to classify whether players will play or not based on climatic condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the result of prediction.

PROBLEM: Players will play if climate is sunny. Is this statement is correct? We can solve it using above discussed method of posterior probability.

P (Yes | Sunny) = P (Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes) = 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of various class based on different attributes. This algorithm is mostly used in text classification and with issues having multiple classes.

VARIATIONS OF THE NAIVE BAYES ALGORITHM

Here, scikit learn (python library) will help here to construct a Naive Bayes model in Python. There are three types of Naive Bayes model under scikit learn library:

1.GAUSSIAN: It is used in classification and it assumes that features follow a normal distribution.

$$P(x_j|C_i) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}}\exp\left(-\frac{(x_j-\mu_{C_j})^2}{2\sigma_{C_i}^2}\right)$$

2.MULTINOMIAL: It implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically

ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

represented as word vector counts, although tf-idf vectors are too known to work well in practice). The distribution is parametrized by vectors θy= (θy1, ……θyn) for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability P(xi |y) of feature i appearing in a sample belonging to class y.

The parameters θy is estimated by a smoothed version of maximum probability, i.e. relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where Nyi=∑x∈Txi is the number of times feature i appears in a sample of class y in the training set T, and Ny=∑i=1nNyi is the total count of all features for class y.

The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

3.BERNOULLI: It implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable .In this manner, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).

The decision rule for Bernoulli naive Bayes is based on

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

which differs from multinomial NB's rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature.

In the case of text classification, word occurrence vectors (rather than word count vectors) may be used to train and use this classifier. BernoulliNB might perform better on some datasets, especially those with shorter documents. It is advisable to evaluate both models, if time permits.

Spark has a built-in function called the CrossValidator to conduct cross validation which begins by splitting the training dataset into a set of "folds" which are used as separate training and test datasets. For example, with k=5 folds, CrossValidator will generate 5 different (training, test) dataset pairs, each of which uses 4/5 of the data for training and 1/5 for testing. To evaluate a particular Parameter (specified in the paramgrid), CrossValidator computes the average evaluation metric for the 5 Models produced by fitting the Estimator on the 5 different (training, test) dataset pairs and tells you which model performed the best once it is finished.

After identifying the best ParamMap, CrossValidator finally re-fits the Estimator using the best ParamMap and the entire dataset.

Low heart rate causes a risk of death, heart disease, and cardiovascular diseases. Therefore, monitoring the heart rate is critical because of the heart's function to discover its irregularity to detect the health problems early. Rapid technological advancement (e.g., artificial intelligence and stream processing technologies) allows

ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

healthcare sectors to consolidate and analyze massive health-based data to discover risks by making more accurate predictions. Therefore, this work proposes a real-time prediction system for heart rate, which helps the medical care providers and patients avoid heart rate risk in real time. The proposed system consists of two phases, namely, an offline phase and an online phase. The offline phase targets developing the model using different forecasting techniques to find the lowest root mean square error. The heart rate time-series dataset is extracted from Medical Information Mart for Intensive Care (MIMIC-II). Recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent units (GRU), and bidirectional long short-term memory (BI-LSTM) are applied to heart rate time series. For the online phase, Apache Kafka and Apache Spark have been used to predict the heart rate in advance based on the best developed model. According to the experimental results, the GRU with three layers has recorded the best performance. Consequently, GRU with three layers has been used to predict heart rate 5 minutes in advance.

REFERENCES

1. Ali, Farman, et al. "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion." *Information Fusion* 63 (2020): 208-222.
2. Shafi, Jana, et al. "Prediction of heart abnormalities using deep learning model and wearabledevices in smart health homes." *Multimedia Tools and Applications* (2021): 1-15.
3. Venkatesh, R., C. Balasubramanian, and MadasamyKaliappan. "Development of big data predictive analytics model for disease prediction using machine learning technique." *Journal of medical systems* 43.8 (2019): 1-8.
4. Saleh, Hager, et al. "Predicting systolic blood pressure in real-time using streaming data and deep learning." *Mobile Networks and Applications* 26.1 (2021): 326-335.
5. Faker, Osama, and ErdoganDogdu. "Intrusion detection using big data and deep learning techniques." *Proceedings of the 2019 ACM Southeast Conference*. 2019.