European Journal of Emerging Technology and Discoveries
ISSN (E): 2938-3617
Volume 1, Issue 8, November, 2023.

ALGORITHM OF NAIVE BAYES METHODS IN BINARY CLASSIFICATION TASKS ON SANTANDER DATASET EXAMPLE FROM KAGGLE PLATFORM

1 Utemuratov Rustam Bazarbaevich ,
2 Ollamberganov Fayzulla Farxod o'g'li ,
3 Pirjanov Nurlan Baxtiyarovich ,
1,2,3 Karakalpak state university after named Berdakh
2 fayzulla@gmail.com

Abstract
The actual output of many binary classification algorithms is a prediction score. The score indicates the system's certainty that the given observation belongs to the positive class. Naive Bayes is among the simplest probabilistic classifiers. It often performs surprisingly well in many real-world applications, despite the strong assumption that all features are conditionally independent given the class. In the learning process of this classifier with the known structure, class probabilities and conditional probabilities are calculated using training data, and then values of these probabilities are used to classify new observations. To make the decision about whether the observation should be classified as positive or negative, you will interpret the score by picking a classification threshold (cut-off) and compare the score against it. Any observations with scores higher than the threshold are then predicted as the positive class and scores lower than the threshold are predicted as the negative class. However, AUC is independent of the selected threshold, you can get a sense of the prediction performance of your model from the AUC metric without picking a threshold

Keywords: Naive Bayesian Classifier, Gaussian mixture model, Kernel Naive Bayes, machine learning, data classification.

Introduction

The paper considers three models for a Bayesian classifier: a Gaussian naive Bayes classifier, a Bayesian Gaussian mixture model, and a naive Bayes with distribution density estimation. Numerical experiments are carried out on a real Santander binary classification dataset taken from the Kaggle platform, where continuous features are discretized by applying the listed methods. The performance characteristics of these models are compared with a fully connected neural network and libraries that implement gradient boosting algorithms LightGBM , XGBoost and CatBoost [1]. The results demonstrate that the proposed models can improve the performance of the Naive Bayes classifier and compete with popular boosting algorithms.
Experiments with Santander data. The data was used in the «Santander Customer Transaction Prediction» on the Kaggle platform. This competition involves determining which customers will make a specific transaction in the future, regardless of the transaction amount. The goal of the competition was to build the most efficient classifier based on an anonymized dataset containing numeric continuous features and a binary target column. The task was to predict the value of the target column in the test set. The data corresponds to 400,000 records, of which 200,000 with unknown labels were allocated for testing and summing up the results of the

competition. Transaction data has a structure formalized from 200 continuous attributes var0-var199.

Let's apply Bayesian inference to Santander customer transaction data that has a binary target variable and 200 continuous factor functions. We model the target variable as an unknown vector Y, and the features as a matrix X. The prior probability $p_Y(y)$reflects knowledge before observation. In this problem, the quantity Y has a discrete Bernoulli distribution (only two classes), which can be determined by specifying a positive probability - the proportion of the positive class in the data. Probability $f_{X|Y}(x,y)$models the distribution of an observation, taking into account the familiarity of class labels. Posterior probability $p_{X|Y}(y,x)$is updated knowledge about an unknown target variable after an observation. The MAP (Maximum A Posteriori) estimation method selects the class with the highest posterior probability. For binary classification, this has the same effect as setting the threshold to 0.5 for a positive posterior probability. LSM (least squares method) E[Y|X] selects the mean of the posterior distribution. For binary classification, this is simply the positive posterior probability $p_{X|Y}(1,x)$that needs to be predicted. Bayes' rule for this problem is

$$p_{Y|X}(y|x) = \frac{p_Y(y)f_{X|Y}(x|y)}{\sum_{y'}^{n} p_Y(y')f_{X|Y}(x|y')} \quad (1)$$

Here Xrepresents a sequence of 200 observations $X_0, X_1, \dots, X_{199}$.

Thus, $p(y = 1)$is the prior probability of a positive class. And $p(y = 0)$it can also be easily calculated as $1 - p(y = 1)$. The problem is how to calculate the 200 other terms, that is, how to calculate the following: $p(x_i|y = 1)$, as well as $p(x_i|y = 0)$. There are two main ways to calculate this. The first way to do this is to assume that the i-th feature (x_i)follows a Gaussian distribution, such as a normal distribution, and calculate $p(x_i \mid y = 1)$from the probability density function (PDF) of the normal distribution.

However, when conducting exploratory analysis, it can be clearly seen that not all 200 features follow a Gaussian distribution. Thus, assuming a Gaussian distribution may not be the best choice for estimating $p(x_i|y)$. However, this model is very simple and effective - it is worth considering it and using the main ideas further.

We assume that the probability distributions are normal and independent. From this we get a Gaussian Naive Bayes classifier (Gaussian means normal, and naive means independent):

$$p_{Y|X_0,X_1\dots,X_{im}}(y \mid x_0, x_1, \dots, x_{199}) = \frac{p_Y(y)\prod_{i=0}^{199} f_{Xi|Y}(x_i|y)}{\sum_{y'=0}^{1} p_Y(y')\prod_{i=0}^{199} f_{Xi|Y}(x_i|y')} \quad (2)$$

The classifier is already implemented in the scikit-learn library [1], so we can use it right away. Trait distributions have different mean and standard deviation, so it is worth standardizing them so that they have zero mean and unit variance. In addition, some feature distributions have slight jaggies on the left or right. You can use a quantile transform to remove small jaggies. In the practice of statistical research, the concept of ROC curve is used for the quality of binary classification (Receiver Operation Characteristic) [2]. The higher the monotonically increasing ROC curve goes, the better the classification quality. AUC Score (Area Under Curve) represents the area under the ROC curve, and the closer the AUC is to one, the better the classification score. In practice, we may skip the quantile transformation. It turns out that this conversion provides only a minor improvement performance (0.001 AUC on cross-validation), while requiring significantly more computation. When evaluating the model using the cross-

validation method over five blocks, we obtain AUC = 0.889, while complex customized boosting algorithms [3] give results of about 0.902 for AUC. In practice, this is a minor difference.

To implement naive Bayes with a distribution density estimate, we introduce a kernel density estimate to calculate the probability density of an arbitrary feature distribution. This method discards the assumption that all features are normally distributed. On cross-validation, we also get a very impressive result AUC = 0.895, which illustrates the advantage of a naive Bayes classifier with distribution density estimation.

Now we will try to improve the Gaussian Naive Bayes classifier again by replacing the Gaussian model with a more flexible Gaussian mixture model. The posterior probability $p_Y(y)$will be taken as the ratio of the two classes, and the probability $f_{X_i|Y}(x_i|y)$will be obtained by fitting the data to a Gaussian mixture model.

The Gaussian mixture model produces a mixture of normal distributions. We can use sklearn.mixture.GaussianMixture [1] to fit the data. It is also necessary to standardize features; data with different means and variances may degrade the ability to train a Gaussian mixture model. There are two important hyperparameters: n_components – is the number of normal distributions to mix, and reg_covar – is a regularization parameter.

We will use the Gaussian mixture model to estimate the probability density function $f_{X_i|Y}(x_i \mid y)$. Since multiplying a large number of small numbers will lead to overflow, we take the logarithm of expression (2) and convert the products into sums:

$$\ln\, p_{Y|x_0,x_1,\dots x_{199}}(y|x_0,|x_1,\dots,x_{199}) = \ln\, p_Y(y) + \sum_{i=0}^{199} \ln\, f_{(x_i|Y)}(x_i \mid y) - \ln \sum_{y'=0}^{1} \exp\left(\ln p_Y(y')|\sum_{i=0}^{199} \ln f_{(X_i|Y)}(x_i|y)\right) \quad (3)$$

To determine the probabilities, it is necessary to perform an inverse transformation in formula (3). Naive Bayesian Gaussian mixture shows an AUC improvement of up to 0.899 compared to Gaussian Naive Bayesian, although it takes a little longer to train. The advantage of this method is that, like the density estimation method, it is more flexible and does not require the data to be normally distributed. The signs are also conditionally independent.

Whatever method researchers use, the goal is to have a model that is simple, easy to compute, and accurate (describes the real data very well).

References

1. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018. — 576 с.
2. Hastie T., Tibshirani R., Friedman J.H. The Elements of Statistical Learning, 2nd. – Springer, 2009. – 533 p.
3. Никулин В.Н., Палешева С.А., Зубарева В.С. Об однородных ансамблях при использовании метода бустинга в приложении к несбалансированным данным // Вестник Пермского университета. Сер. Экономика. 2012. №1. С. 7-14.
4. BAYHONOV, B., & UTEMURATOV, R. MAIN DIRECTIONS OF DEVELOPMENT OF SMALL BUSINESS AND PRIVATE ENTREPRENEURSHIP IN THE REPUBLIC OF UZBEKISTAN. ЭКОНОМИКА, (2), 611-615.

5. Utemuratov, R. B. (2021). ANALYSIS OF EFFICIENCY OF IMPLEMENTATION OF INVESTMENT PROJECTS IN THE REPUBLIC OF KARAKALPAKSTAN. International journal of Business, Management and Accounting, 1(3).
6. Турганбаев, А., Олламберганов, Ф., & Калбаев, А. (2023). АЛГОРИТМЫ АВТОМАТИЧЕСКОГО АНАЛИЗА ДОКУМЕНТА. Talqin va tadqiqotlar, 1(22).
7. Kalbaev, A. M., & ulı Turganbaev, A. J. (2023). HUJJATLARDAGI BELGILARNI TANIB OLISH UCHUN ILG ‘OR AI TEXNOLOGIYALARIDAN FOYDALANISH. Innovative Development in Educational Activities, 2(2), 156-158.
8. Kalbaev, A. M., & ulı Turganbaev, A. J. (2022, December). HUJJATLARNI TANIB OLISHDA ILG ‘OR SUN’IY INTELLEKT TEXNOLOGIYALARINI QO ‘LLASH. In INTERNATIONAL CONFERENCE DEDICATED TO THE ROLE AND IMPORTANCE OF INNOVATIVE EDUCATION IN THE 21ST CENTURY (Vol. 1, No. 9, pp. 3-6).