A mathematical essay on Naive Bayes Classifier

Assignment 3 : EE4708 Data Analytics Laboratory

Madhur Jindal
Inter Disciplinary Dual Degree Program in Data Science
Indian Institute of Technology (IIT) Madras
Chennai, India
me18b059@smail.iitm.ac.in

***Abstract*—This document is a mathematical essay on Naive Bayes Classifier wherein it is applied on a dataset containing different attributes from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). The key task is to determine whether a person makes over 50K a year, from their age, workclass, education, marital status, occupation etc, and determine whether some groups of people are more likely to have an income higher than 50K per month.**

I. Introduction

Classification is the process of learning a mapping from related input features to discrete data classes or categories. In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models,[1] but coupled with kernel density estimation, they can achieve higher accuracy levels.Another assumption made here is that all the predictors have an equal effect on the outcome. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Naïve Bayes models are also known as simple Bayes or independent Bayes. All these names refer to the application of Bayes' theorem in the classifier's decision rule. Naïve Bayes classifier applies the Bayes' theorem in practice. This classifier brings the power of Bayes' theorem to machine learning.

II. Naive Bayes Classifier

A. Different Types

1) **Multinomial Naive Bayes**
With a Multinomial Naïve Bayes model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, . . . ,pn) where pi is the probability that event i occurs. Multinomial Naïve Bayes algorithm is preferred to use on data that is multinomially distributed. It is one of the standard algorithms which is used in text categorization classification.
2) **Bernoulli Naive Bayes**
n the multivariate Bernoulli event model, features are independent boolean variables (binary variables) describing inputs. Just like the multinomial model, this model is also popular for document classification tasks where binary term occurrence features are used rather than term frequencies.
3) **Gaussian Naive Bayes**
When we have continuous attribute values, we made an assumption that the values associated with each class are distributed according to Gaussian or Normal distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μi be the mean of the values and let σi be the variance of the values associated with the ith class. Suppose we have some observation value xi . Then, the probability distribution of xi given a class can be computed by the following equation –

$$P\left(x_i \mid y\right) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

B. Model structure

Naïve Bayes Classifier uses the Bayes' theorem to predict membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as the Maximum A Posteriori (MAP).

The MAP for a hypothesis with 2 events A and B is

MAP (A)

$$= max(P(A|B)) \tag{1}$$

$$= max\frac{(P(B|A) * P(A))}{P(B)} \tag{2}$$

$$= max(P(B|A) * P(A)) \tag{3}$$

Here, P (B) is evidence probability. It is used to normalize the result. It remains the same, So, removing it would not affect the result.

Identify applicable funding agency here. If none, delete this.

Naïve Bayes Classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

In real world datasets, we test a hypothesis given multiple evidence on features. So, the calculations become quite complicated. To simplify the work, the feature independence approach is used to uncouple multiple evidence and treat each as an independent one.

C. Metrics for model evaluation - Confusion Matrix

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in the dataset. It gives you insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.

Prediction outcome

actual value	**p**	**n**	**total**
p′	True Positive	False Negative	P′
n′	False Positive	True Negative	N′
total	P	N	

we can assign the event row as “positive” and the no-event row as “negative“. We can then assign the event column of predictions as “true” and the no-event as “false“.

This gives us:

“true positive” for correctly predicted event values. “false positive” for incorrectly predicted event values. “true negative” for correctly predicted no-event values. “false negative” for incorrectly predicted no-event values.

1) Precision: Precision can be defined as the percentage of correctly predicted positive outcomes out of all the predicted positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true and false positives (TP + FP).

So, Precision identifies the proportion of correctly predicted positive outcome. It is more concerned with the positive class than the negative class.

Mathematically, precision can be defined as the ratio of TP to (TP + FP).

2) Recall: Recall can be defined as the percentage of correctly predicted positive outcomes out of all the actual positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). Recall is also called Sensitivity.

Recall identifies the proportion of correctly predicted actual positives.

Mathematically, recall can be given as the ratio of TP to (TP + FN).

3) F1-Score: f1-score is the weighted harmonic mean of precision and recall. The best possible f1-score would be 1.0 and the worst would be 0.0. f1-score is the harmonic mean of precision and recall. So, f1-score is always lower than accuracy measures as they embed precision and recall into their computation. The weighted average of f1-score should be used to compare classifier models, not global accuracy.

4) Support: Support is the actual number of occurrences of the class in our dataset.

5) Null-Hypothesis and P-value: The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (¡ 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor’s value are related to changes in the response variable.

Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.

D. ROC AUC

1) ROC Curve: Another tool to measure the classification model performance visually is ROC Curve. ROC Curve stands for Receiver Operating Characteristic Curve. An ROC Curve is a plot which shows the performance of a classification model at various classification threshold levels.

The ROC Curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels.

True Positive Rate (TPR) is also called Recall. It is defined as the ratio of TP to (TP + FN).

False Positive Rate (FPR) is defined as the ratio of FP to (FP + TN).

In the ROC Curve, we will focus on the TPR (True Positive Rate) and FPR (False Positive Rate) of a single point. This will give us the general performance of the ROC curve which consists of the TPR and FPR at various threshold levels. So, an ROC Curve plots TPR vs FPR at different classification threshold levels. If we lower the threshold levels, it may result in more items being classified as positve. It will increase both True Positives (TP) and False Positives (FP).ROC curve help us to choose a threshold level that balances sensitivity and specificity for a particular context.

2) ROC AUC: ROC AUC stands for Receiver Operating Characteristic - Area Under Curve. It is a technique to compare classifier performance. In this technique, we measure the area under the curve (AUC). A perfect classifier will have a ROC AUC equal to 1, whereas a purely random classifier will have a ROC AUC equal to 0.5.

So, ROC AUC is the percentage of the ROC plot that is underneath the curve.OC AUC is a single number summary of classifier performance. The higher the value, the better the classifier.

ROC AUC of our model approaches towards 1. So, we can conclude that our classifier does a good job in predicting whether it will rain tomorrow or not.

III. Problem - Understanding and Modelling

We are presented with a problem involving census data of certain given attributes of the people of various countries including Sex, Age, education, Marital Status, Occupation, Relationship, Finalwgt, Race, Captial gain & loss, working hours per week, workclass and their native country. Using all these features, we are to predict whether the given person makes over 50K per year or not, and also identify relations between the features and the target class.

A. Data Pre-processing

We start with reading the data to a pandas dataframe. We observe a total of 32561 data points, with 15 columns in total, including the different features related to the person. On visualizing the distributions of the people survived we see that around 24.1% of the total people have a higher income than 50K. (Figure 1) Out of the total 15 features, 9 are categorical and 6 are numerical. We then move on to checking for the null values in the data and find that we do not have any NaN values, but three of the columns Workclass, Occupation and native country have '?' marks as some data points, which need to be treated. The first step is to replace these '?' marks with NaN values and then finally imputing them with the mode of each of the columns as the number of null values is very low and thus no special treatment is necessary.

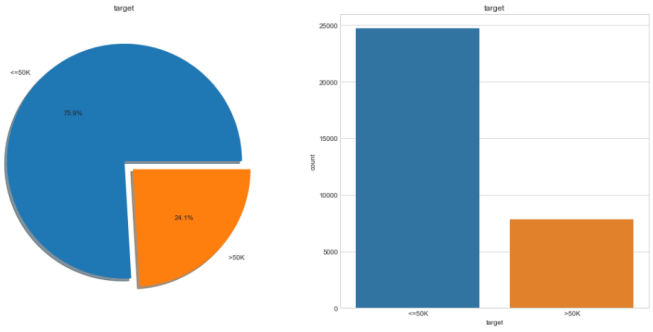

Fig. 1. Target Distribution

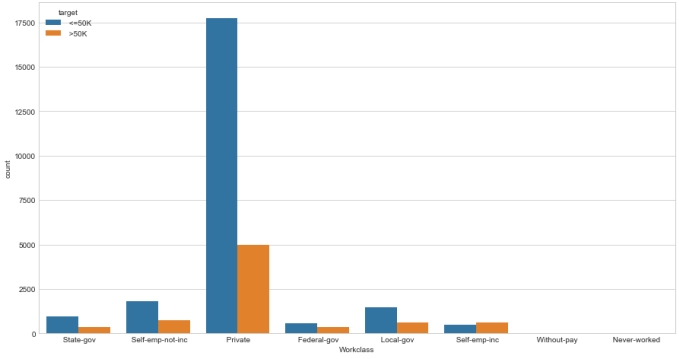

Fig. 2. Count-plot Work Class

We then check for the cardinality of each of the categorical features, that is the number of different unique values each feature can take as a higher number can cause problems, we find that most of the features do not have more than 7 attributes, wherein the native country feature has the highest number.

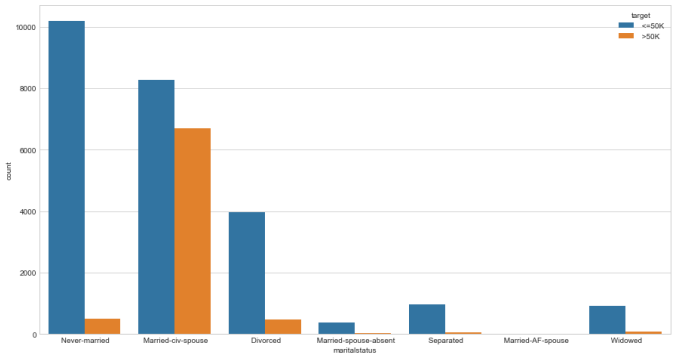

Fig. 3. Count-plot Marital Status

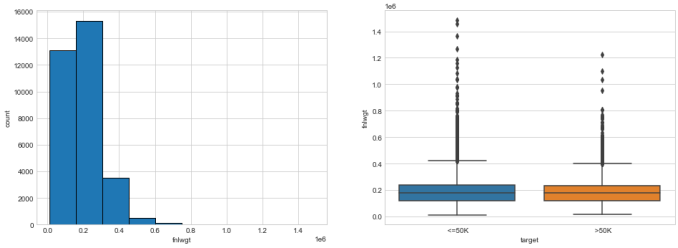

Fig. 4. FnlWgt Distribution

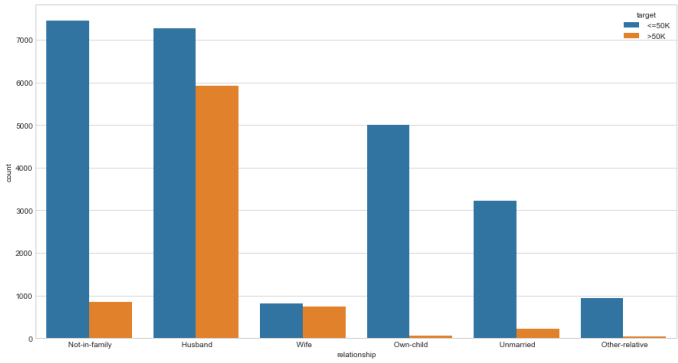

Fig. 5. Count-Plot Relationship

B. Visualization and Feature Generation

Now we move on to visualizing each of the features one by one. Starting with workclass, we see that the main types are government employees, Private, Self employed, without pay and never worked classes, with the rate of higher income being different for each of the classes, with the highest rate for self employed inclusive, and the least for private. Moving on to education, the main classes are university level, school level, and postgraduate level, with the trend being lesser income for

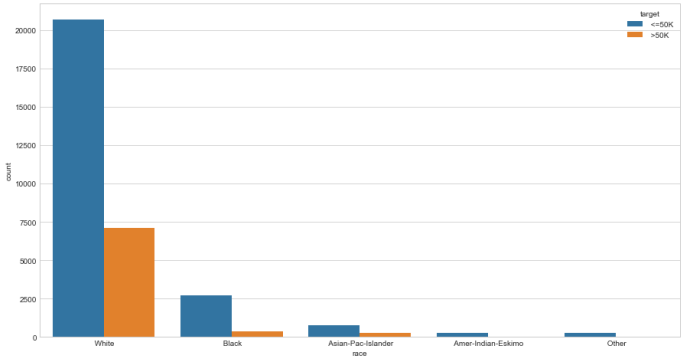

Fig. 6. Countplot Race

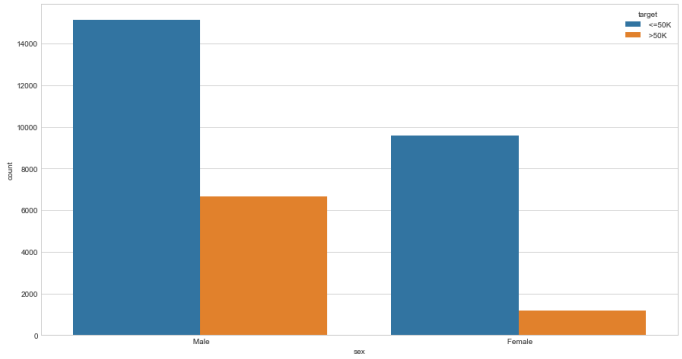

Fig. 7. Countplot - Sex

lesser level and the ones with doctorate and professional school enjoying the highest of salaries.

Moving on to marital status, we see that people married and with spouse enjoy the highest of incomes, while all the others having very low possibilities of high income. Checking occupation, we see that the number of classes increases tremendously, with each class enjoying different rates, with Executive managers and professional speciality enjoying the highest incomes. Following this, we move on to relationship feature, and find out that husbands and wifes have the highest probability of having high income, as evident in the society, while single people without family or unmarried do not have such high incomes. Moving on to race, we see a bias towards white people having higher salaries, as compared to the others. A yet disturbing fact is seen when sex is visualized, with males enjoying more income on average than females.

On exploring Age using histograms and boxplots, we see that people of the higher age in general are seen to have higher incomes, which is explainable as most people are not paid well in the starting of their careers. Using this information, we create a new variable isChild that specifies if the person is less than 24 years of age, and we see that children have zero chance of higher income, from the given data. Post this, we also create a feature senior citizen, which is ticked after being 50 years of age or more. We see that senior citizens have more chance of having higher incomes.

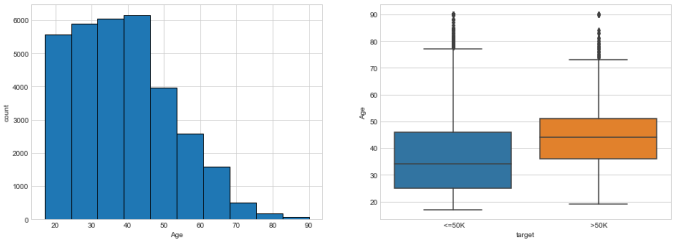

Fig. 8. Age Distribution

We then move forward to education-num which specifies the number of education levels, and see a trend that higher income is captured by people with higher education levels, owing to which we create a new feature highly educated. Similarly, we also explore hourspw which is number of hours worked per week, and see that the people with higher working hours usually tend to have higher incomes, thus creating a new variable work more.

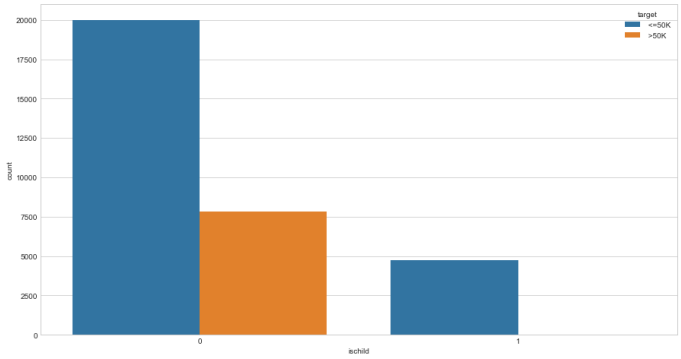

Fig. 9. Countplot Children

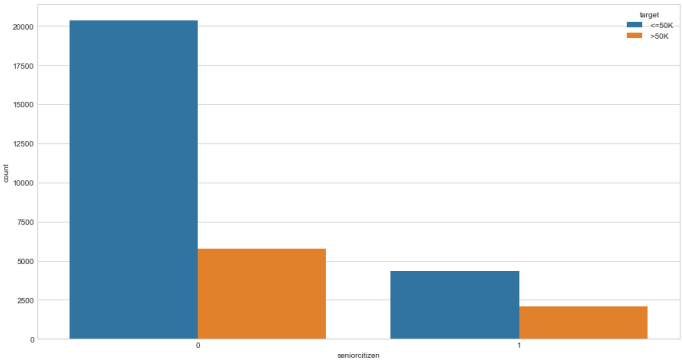

Fig. 10. Countplot Senior Citizen

Finally as part of postprocessing, we create dummies for the categorical variables so that they are meaningful to the machine, which is also termed as One Hot encoding.

C. Guassian Naive Bayes Classifier Modelling

We now move on to modelling using the sklearn library's implementation of the gaussian naive bayes classifier. The

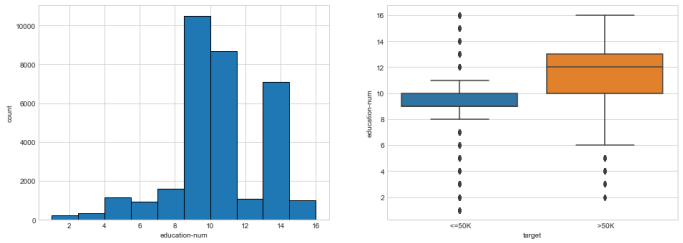

Fig. 11. Education Number distribution

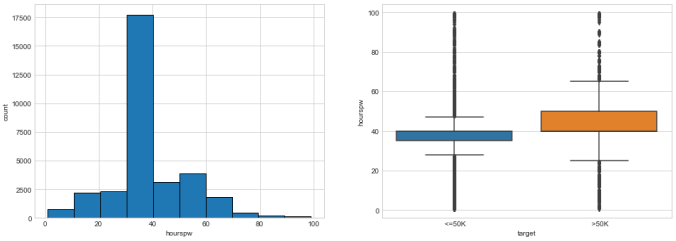

Fig. 13. Hours per week distribution

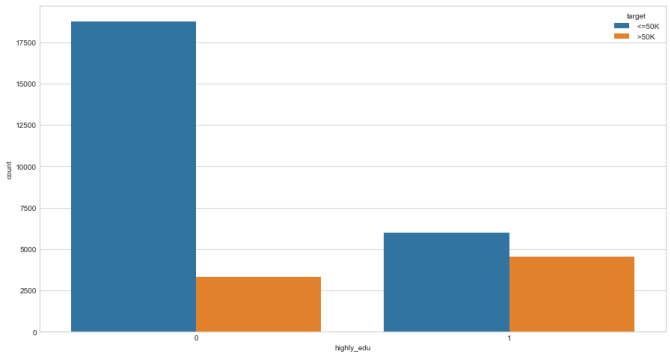

Fig. 12. Countplot Highly Educated

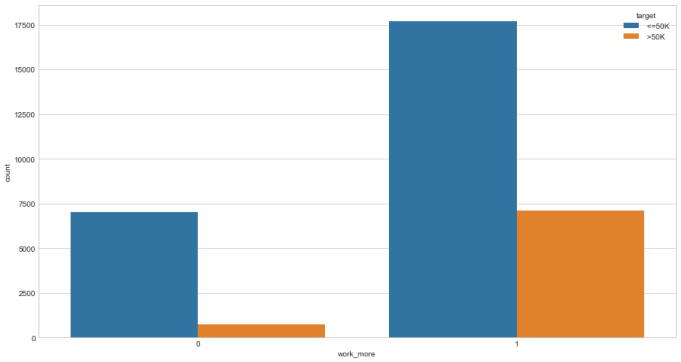

Fig. 14. Count-Plot Working More

first step is to divide the data into a train and a validation set, so that we could test on unseen data. The next step is to scale the features for which we use sklearn's robust scaler. The next task is to get a model with the right set of hyperparameters, which is decided by using randomized search croos validation technique from sklearn's model selection toolkit. The hyperparameter being fine-tuned is var smoothing in the logarithmic scale. Thus training and predicting using the tuned model, we get an accuracy of 82.6 percent on the train set and 82.8 percent on the test dataset. We see that the accuracy values for test and train are close, thus no sign of overfitting.

Then we proceed to checking for the evaluation metrics and start with the Confusion Matrix, we observe that there are a total of 6604 true positives, 1482 true negatives, 851 false positives and 832 false negatives. Moving on to the classification report we observe an f1 score of 0.89 for ¡= 50K anf 0.64 for ¿50K, with an accuracy of 0.83, macro average of 0.76 and weighted average of 0.83. We see that these are good values and our model is performing well.

Following this, we plot the ROC curve, which is basically false positive rate vs true positive rate, and we see that the curve is well above the y = x line, which is a good indication with an AUC value of 0.8843.Following this we move on to test for the variability in performance on testing and training with different datasets, using 10-fold cross validation. We see that the mean accuracy is close to the original one, and also there is not much deviation from the average for all the folds, thus we can say our model is not much reliant on the data on which it is being trained. Finally we move on to test different thresholds to get the best accuracy on the test set, and find that 0.8 threshold gives us the best accuracy of 0.835.

IV. Conclusion

Having done a thorough analysis, we see that our gaussian naive bayes classifier is performing well on the dataset, as a baseline model always predicting the major class will give us an accuracy of 0.76 and our model gives an accuracy of 0.836. So, we can conclude that our Gaussian Naïve Bayes classifier model is doing a very good job in predicting the class labels. ROC AUC of our model approaches towards 1. So, we can conclude that our classifier does a very good job in predicting whether a person makes over 50K a year. If we look at all the 10 scores produced by the 10-fold cross-validation, we can also conclude that there is a relatively small variance in the accuracy between folds, ranging from 83.68 percent accuracy to 81.18 percent accuracy. So, we can conclude that the model is independent of the particular folds used for training.

Some observations from the data were that some classes of people were likely to have more income such as, males than females, white people than others. We also see that young people are less likely than older people to enjoy high income, which is explainable as most people are not paidwell in the starting of their careers. We also see that people with higher education levels tend to get paid more, and so is the case with number of hours worked per week, more the number of hours, more is the pay. We also see executive managers and professional speciality workers getting the highest pay. Further avenues of growth can be checking for more features that could explain the target variable better and look for minimizing the multicollinearity.

REFERENCES

[1] Classification, Data Analytics Laboratory EE4708, July - November 2021

[2] Niave Bayes Classifier by Prof. Manikandan, Pattern Recognition and Machine Learning, July - November 2021

[3] https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[4] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[5] http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/

[6] https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

[7] https://stackabuse.com/the-naive-bayes-algorithm-in-python-with-scikit-learn/