Predicting Income Class Using a Naive Bayes Classifier

Swathi V
Dept. of Engineering Design
Indian Institute of Technology Madras
Chennai, India
ed18b034@smail.iitm.ac.in

***Abstract*—The annual income of a person can depend on various socio-economic factors, including age, sex, education, occupation, and family status. In this article, we use a Naive Bayes classifier to try and use this socio-economic information to predict the income range of a person. Using this use case as an example, we try to analyse the mathematical foundations of a Naive Bayes classifier, and various aspects related to classification problems such as performance metrics.**

***Index Terms*—Naive Bayes, Bayes Theorem, Decision Rule, Conditional Independence, Income Class**

I. Introduction

Naive Bayes is a simple probabilistic method of modelling classifiers, which can be used for classifying a given data point into one of many finite pre-defined output classes. It has been shown to give good results especially with large datasets, and combined with density estimation techniques, can sometimes outperform more sophisticated models.

A Naive Bayes classifier uses class conditional probabilities found using the Bayes Theorem to classify points using the Bayes Decision Rule. These classifiers are built on the assumption that each of the features affecting the output are conditionally independent, i.e., within each class, the features are independent. Though this assumption does not hold true in many real world problems, it is a good approximation in many cases. This assumption also helps in reducing the model complexity.

In this article, the aim is to build a Naive Bayes classifier to predict the annual income class ($>= 50K$ or $< 50K$) of the US population. We try to understand the effect of various causal factors such as age, sex, education, and occupation in determining the same. We also try to understand how in spite of it's strong assumptions, the Naive Bayes model can efficiently be used in such classification problems.

We start with a mathematical overview of the Naive Bayes classifier and discuss it's implementation. We then try to analyse the data visually and try to gain insights about which factors influence the output to a large extent. We use these insights to build a classification model, and finally quantify the performance of the model.

II. Naive Bayes Classifier

A Naive Bayes classifier is a probabilistic approach that is especially useful in situations where the outcome of a classification problem is to be determined by the probabilities of various causal factors. While modelling such a classifier, we make the assumption that the decision problem is posed only in probabilistic terms, and that all the relevant probabilities are known. The building blocks of a Naive Bayes Classifier are the following - Bayes theorem and Bayes Decision Rule. In this section, we analyse each of these separately. We then go on to define a Naive Bayes classifier, and the assumptions made while modelling the same.

A. Bayes Theorem

Let us consider a two class classification problem, with classes w_1 and w_2. Let us assume that we know the **prior probabilities** $P(w_1)$ and $P(w_2)$, as well as the conditional probabilities $P(x|w_i)$ for $i = 1, 2$, which is called the **likelihood** of each class. Then, suppose we observe a new data point x, the probability that is belongs to the class w_i will be given by Bayes Theorem as

$$P(w_i|x) = \frac{P(x|w_i)P(w_i)}{P(x)} \tag{1}$$

where $P(x)$ can be obtained using

$$P(x) = \sum_{i=1}^{2} P(x|w_i)P(w_i) \tag{2}$$

In the above equations, the notation $P(X)$ denotes the probability of event X occurring, and $P(X|Y)$ denotes the probability of event X occurring given that event Y has already occurred. Informally, Bayes Theorem can be expressed as

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

B. Bayes Decision Rule

If we observe a new data point x, then the probability of error while classifying this new data point is

$$P(\text{error}|x) = \begin{cases} P(w_1|x) & \text{if we classify as } w_2 \\ P(w_2|x) & \text{if we classify as } w_1 \end{cases} \tag{3}$$

To minimize this error, we can make the classification decision as

$$\text{Decide } w_1 \text{ if } P(w_1|x) > P(w_2|x); \text{ otherwise decide } w_2 \tag{4}$$

which is called the Bayes Decision Rule.

C. Naive Bayes Classifier

The Naive Bayes Classifier is based on the assumption of **conditional independence** of causal factors, given by

$$P(x_a, x_b|w_i) = P(x_a|w_i)P(x_b|w_i) \tag{5}$$

Using Eq. 1, Eq. 4, and Eq. 5, a Naive Bayes Classifier can be defined as follows.
Let us consider a dataset $X = \{\bar{x}_1, \bar{x}_2, \cdots \bar{x}_n\}$ with feature vectors $\bar{x}_i = [x_{i1},\ x_{i2},\ \cdots\ x_{im}]$, where $x_i \in \mathbb{R}^m$. Let the target variables Y be given by $Y = \{y_1, y_2, \cdots y_n\}$. Since each y_i can belong to p different classes, $y_i \in \{1, 2, \cdots p\}$. Using Bayes Theorem and Conditional independence

$$P(y|x_1, x_2, \cdots x_m) \propto P(y) \prod_{i=0}^{m} P(x_i|y) \tag{6}$$

The assumptions made while modelling a Naive Bayes Classifier are that the causal factors are conditionally independent (for each of the output classes), and that the likelihood and prior are known for every class.

D. Implementation

The *scikit-learn* library in python can be used for implementing a Naive Bayes Classifier, which offers various functions depending on the assumption made about the prior. The major models are explained below.

1) Gaussian Naive Bayes: This model is used when the causal variables are continuous, and are assumed to be gaussian distributed for a given class w_i, i.e.,

$$P(x_i|w_i) = \frac{1}{\sqrt{2\pi\sigma^2_{w_i}}} exp\left(-\frac{(x_i - \mu_{w_i})^2}{2\sigma^2_{w_i}}\right)$$

The parameters σ_{w_i} and μ_{w_i} are obtained using Maximum Likelihood Estimation.

2) Categorical Naive Bayes: This model is used when all the causal factors are discrete and categorically distributed. The

3) Bernoulli Naive Bayes: This model is used when the causal factors are bernoulli distributed, i.e., can take a value of 1 or 0 with probabilities p and $1 - p$ respectively.

4) Multinomial Naive Bayes: A multinomial naive bayes classifier is used when the input features represent some measure of occurance, such as relative frequency. This is used mostly for applications such as Natural Language Processing.

The appropriate model can be selected by visualizing the data and choosing the prior which best matches the causal features.

E. Assessing the Performance of a Model

The performance of a classification model can be quantified using a **confusion matrix** as shown in Fig. 1 and associated scores such as **accuracy**, **precision**, **recall**, and **F1 score**.

The accuracy of a model, which represents the total ratio of correct classifications, is given by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{7}$$

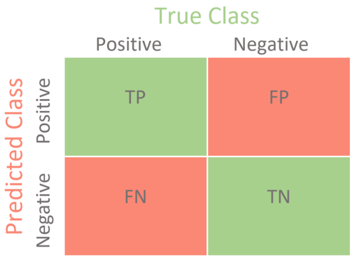


Fig. 1. Confusion matrix

The precision of a model, which represents the ratio of the correct classifications out of all positively predicted values, is given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

The recall of a model, also known as the sensitivity, is the ratio of correct predictions out of all the true positives. It is given by

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

The F1 score combines both precision and recall into a single metric, and is given by

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

The **ROC curve**, which is a plot of the true positive rate vs the false positive rate, can also be used to quantify the performance of a classification model at all classification thresholds. The area under the curve is an aggregate measure of the model across all thresholds, and varies between 0 and 1. A random classifier has an AUC of 0.5, and an ideal classifier has an AUC of 1.

III. Data

The data extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker will be used in this task. The various input features are given in Table I.

IV. The Problem

The aim of this article is to predict the income class of a person based on factors like age, education, occupation, marital status, etc. The steps followed for the same are as follows:

- Data cleaning
- Data visualization and analysis
- Building models and comparing their performances
- Testing the performance on the best model on the test dataset

TABLE I
VARIOUS FEATURES THAT ARE PRESENT IN THE DATASETS

Variable	**Description and Key**
age	Continuous variable
work_class	Can be one of various values - Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
fnlwgt	Final weight, taking into account various factors like population, race, origin - continuous
education	Level of education - Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
education_num	Number of years of education - continuous
marital_status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
race	Categorical variable - White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
sex	Male, Female
capital_gain	Continuous variable
capital_loss	Continuous variable
hours_per_week	Continuous variable
native_country	Categorical
occupation	Can be one of various values - Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces

A. Data Cleaning

A few data points contained missing values (”?”) under the columns *work_class*, *occupation*, and *native_country*. These accounted to 7.3% of the total dataset (2399 out of 32561 entries). For categorical variables, unlike for continuous variables, data imputation is usually done by replacing missing values with the most frequent class. Since in this case this was to be done for a large portion of the data, replacing with the most frequent class will change the distribution of the data significantly. Thus, rows with missing variables were removed.

Apart from rows containing ”?”, all other rows were completely filled.

B. Data Visualization and Analysis

The given dataset contains information about 30162 people. Of these, 7508 (24.8%) belong to the high income class ($> 50K$), and the rest earn $<= 50K$.

The dataset contains 14 variables (6 continuous and 8 categorical). The effect of each of these variables on the income class is visualized separately, as shown in Fig. 2 to 15.

From Fig. 9, we see that the representation of races other than ”White” and ”Black” in the dataset are very minimal. We thus combine all these races into a single bracket, the distribution of which is plotted in Fig. 16. Similarly, it can be seen that the representation of countried other than the United States is minimal, from Fig. 14. All other countries are thus

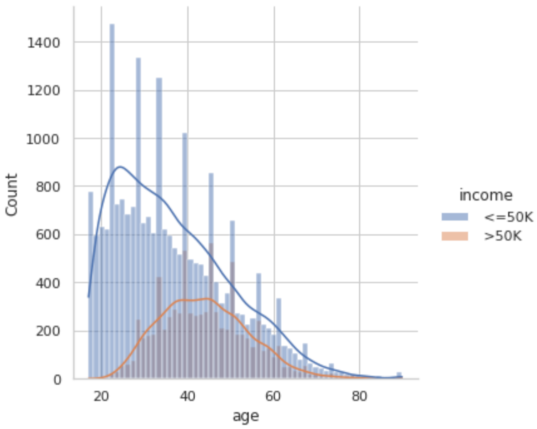


Fig. 2. Distribution of age for each income class

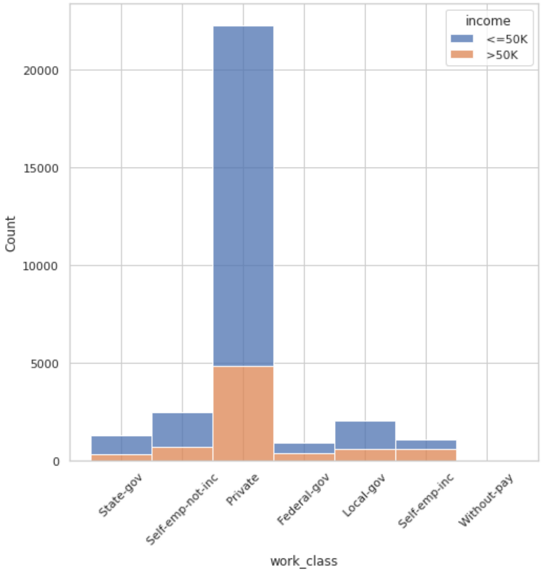


Fig. 3. Distribution of work class for each income class

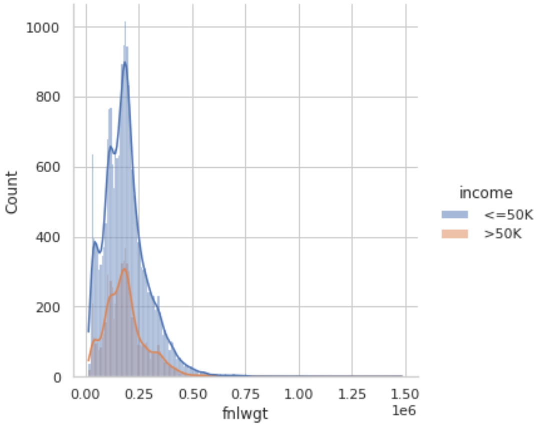


Fig. 4. Distribution of final weight for each income class

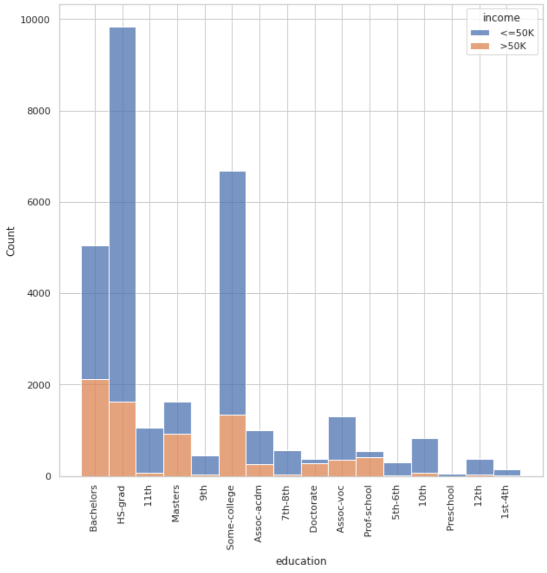


Fig. 5. Distribution of education levels for each income class

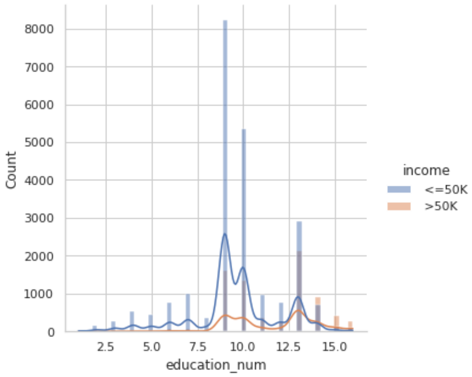


Fig. 6. Distribution of number of years of education for each income class

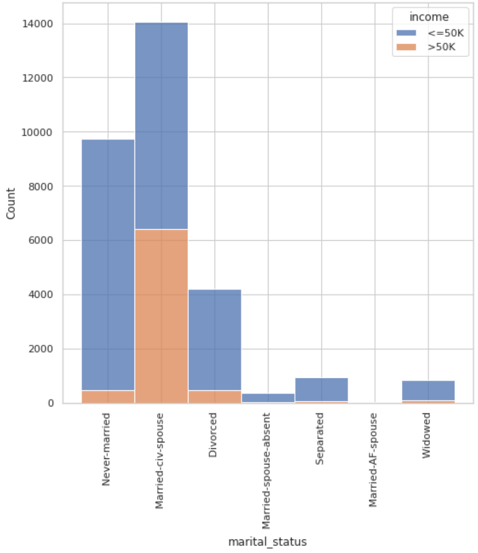


Fig. 7. Distribution of marital status for each income class

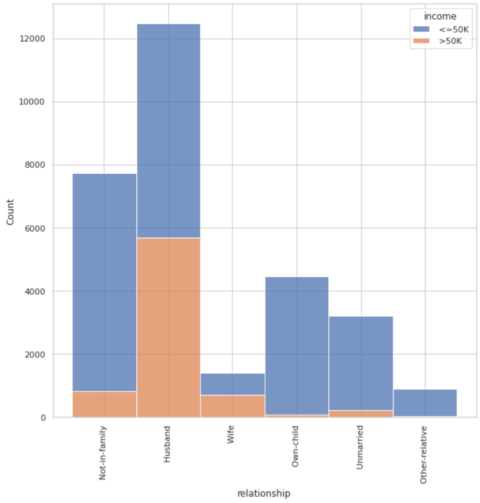


Fig. 8. Distribution of relationship for each income class

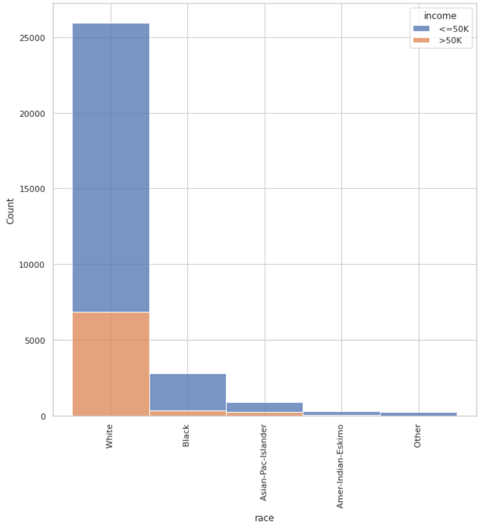


Fig. 9. Distribution of race for each income class

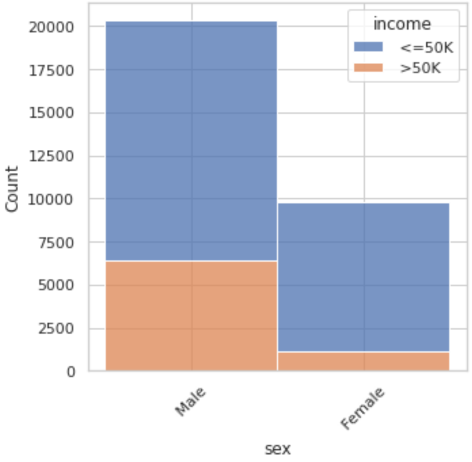


Fig. 10. Distribution of sex for each income class

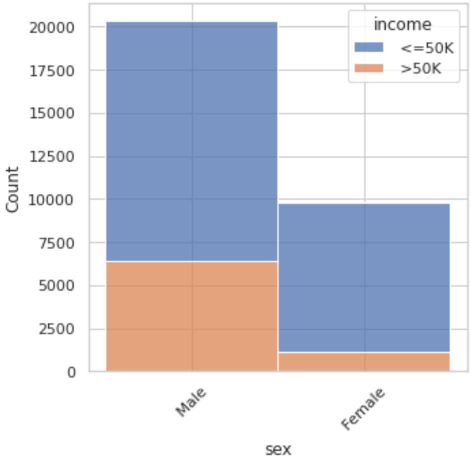


Fig. 11. Distribution of sex for each income class

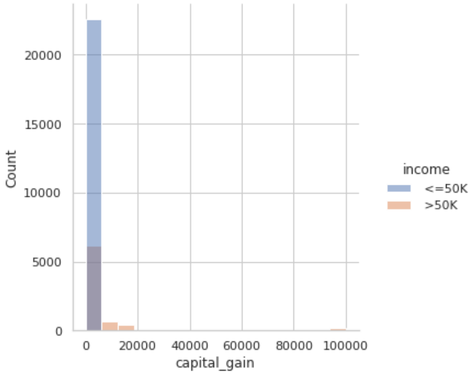


Fig. 12. Distribution of capital gain for each income class

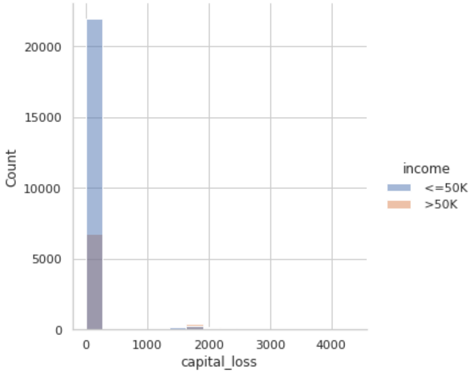


Fig. 13. Distribution of capital loss for each income class

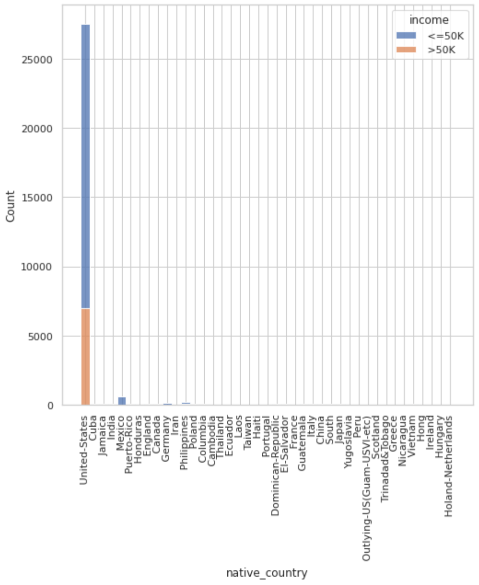


Fig. 14. Distribution of native country for each income class

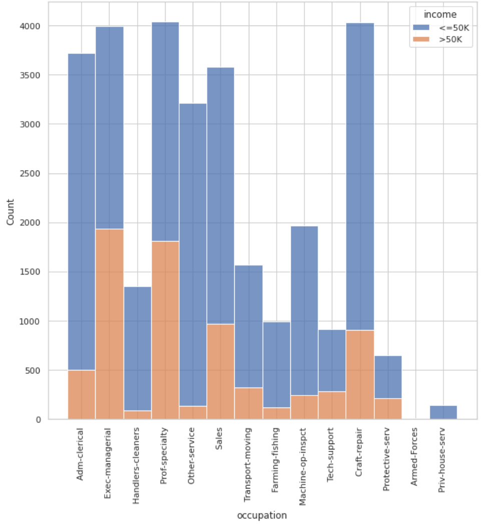


Fig. 15. Distribution of occupation for each income class

combined into a single bracket, and the distribution is plotted in Fig. 17.

From Fig. 12 and Fig. 13, we observe that the variables *capital_gain* and *capital_loss* are very skewed. Hence, these variables are transformed using the *QuantileTransform* function in sklearn. The results are shown in Fig. 18 and 19. We also note that most of the other continuous variables used are not gaussian distributed, but can be roughly approximated to be gaussian.

In order to ensure that features are not redundant, we plot features which are likely to be related against each other to check their distribution. These plots are shown in Fig. 20 to Fig. 23.

From Fig. 20, we see that there is a very strong correlation

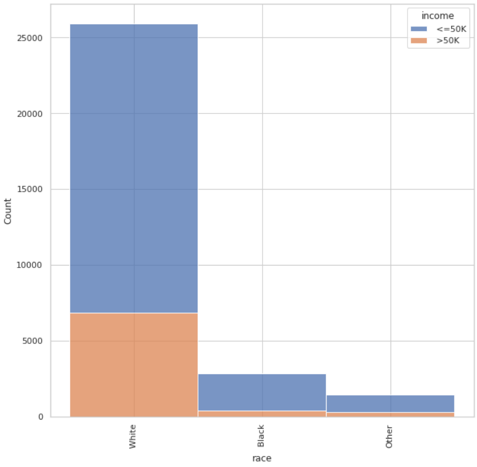


Fig. 16. Distribution of race (modified) for each income class

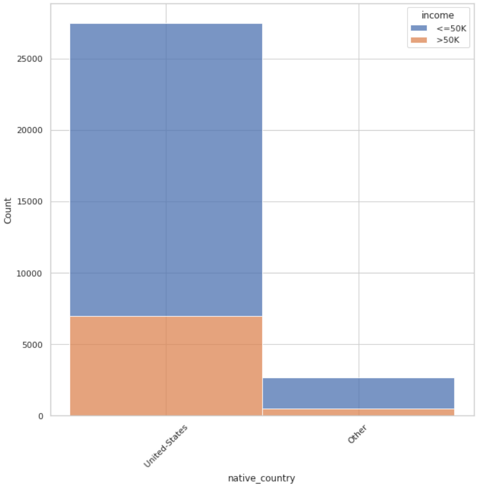


Fig. 17. Distribution of native country (modified) for each income class

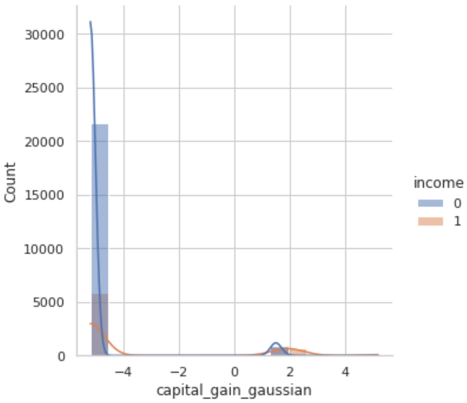


Fig. 18. Quantile transformed capital gain

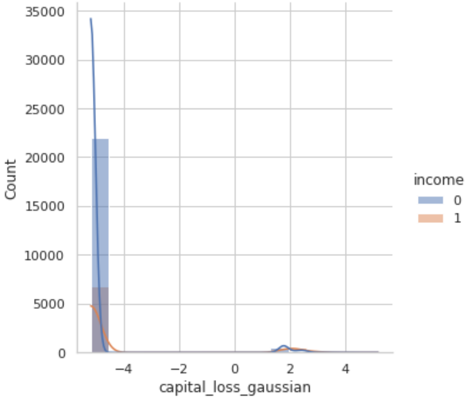


Fig. 19. Quantile transformed capital loss

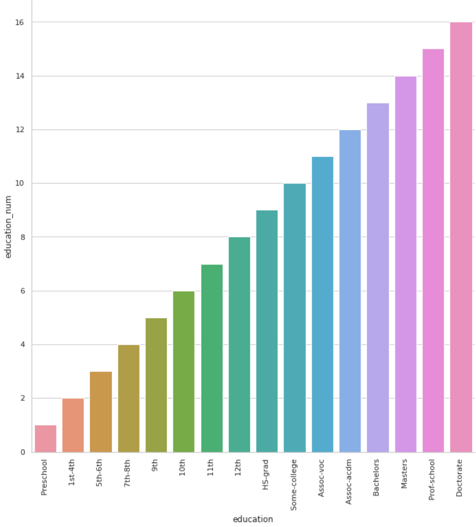


Fig. 20. Variation of education level with number of years

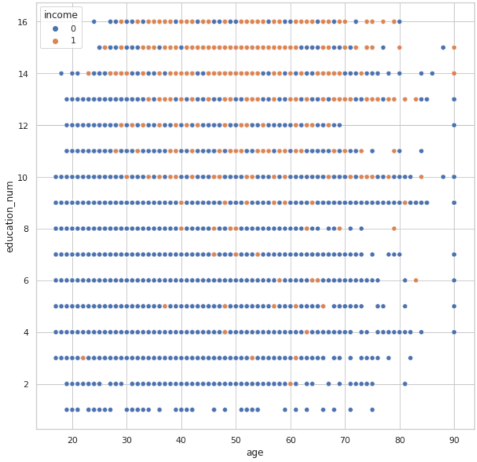


Fig. 21. Relationship between age and number of years of education

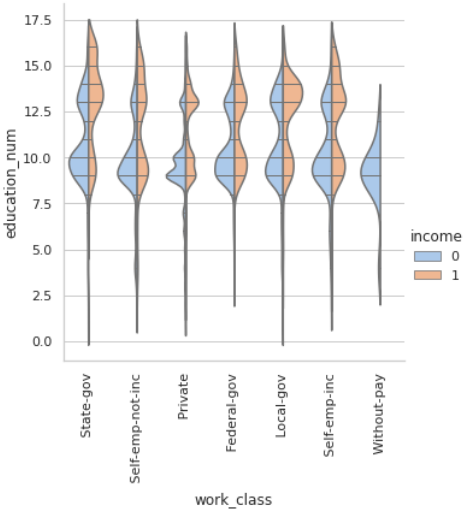


Fig. 22. Relationship between working class and number of years of education

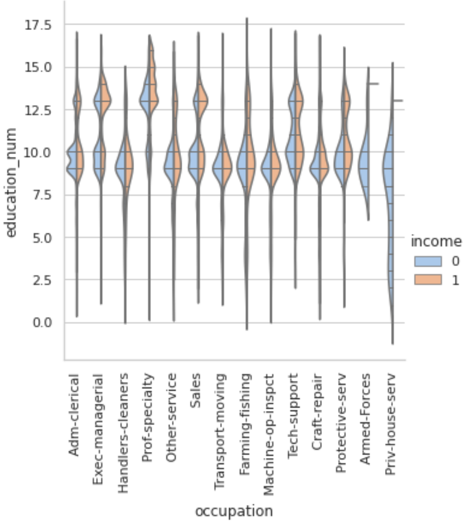


Fig. 23. Relationship between occupation and number of years of education

between the level and number of years of education. Hence, in our analysis, we will only consider the number of years. Fig. 21 to Fig. 23 do not show much correlation between the compared factors. Hence, these variables will all be considered while building the model.

C. Building Models and Comparing Performance

1) Using Categorical Variables: The continuous variables can be binned and converted to categorical variables, which can then be used to train the Naive Bayes model. Binning the *age* variable gives the following classes with counts: $(16.927, 31.6] - 10448$; $(31.6, 46.2] - 11686$; $(46.2, 60.8] - 6222$; $(60.8, 75.4] - 1637$; $(75.4, 90.0] - 169$. Similarly, the variables *fnlwgt* and *hours_per_week* are also binned into five categories each.

The performance of the *CategoricalNB* classifier is shown in Fig. 24.

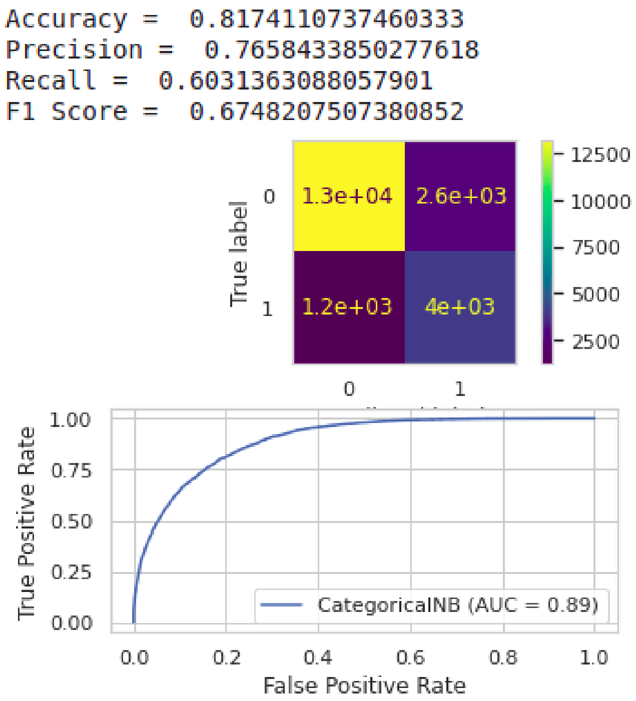

```
Accuracy =  0.8174110737460333
Precision =  0.7658433850277618
Recall =  0.6031363088057901
F1 Score =  0.6748207507380852
```

Fig. 24. Performance of the Categorical Naive Bayes classifier

It can be seen that the model has a good accuracy, precision and roc-auc score, but the recall and f1 score are poor. This is expected because the dataset was skewed, which would have led to better approximation of the probability distribution of one class than the other.

2) Using Normalized Continuous Variables: In this case, all the variables were converted to continuous variables and normalized before being used by the model. The performance of the Gaussian Naive Bayes classifier is given in Fig. 25.

It can be observed that the Gaussian Naive Bayes performs poorer than the Categorical Naive Bayes classifier, which can be explained by the fact that most of the input variables were not Gaussian distributed.

D. Testing the Best Model on the Test Data

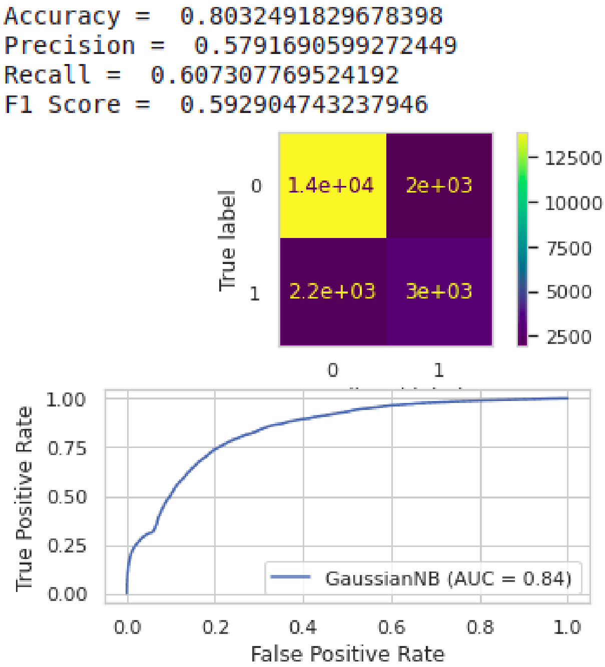
The categorical naive bayes classifier is tested on the test data, the results of which are given in Fig. 26.

The results show that the model does not overfit to the training data.

V. Conclusions

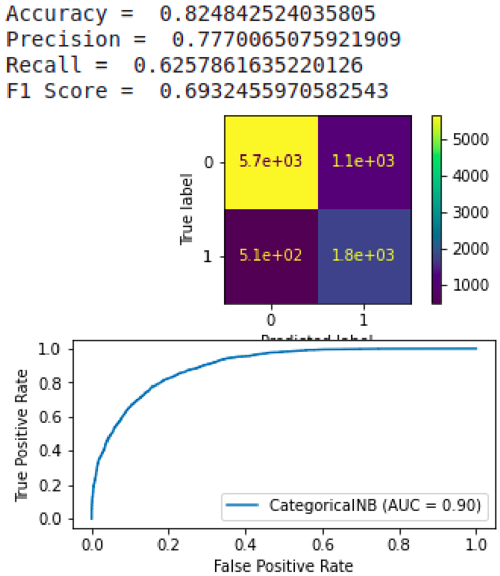
A categorical Naive Bayes classifier provides better results for the given dataset, giving an accuracy of around 80% and an F1 score of around 67% on both the training and test datasets. It is to be noted that even though the dataset did not satisfy all the assumptions of a Naive Bayes Classifier, we were able to modify the data to suit the assumptions roughly.

As future work, methods of improving the recall can be explored. To improve the distribution of the dataset, dataset



```
Accuracy =  0.8032491829678398
Precision =  0.5791690599272449
Recall =  0.607307769524192
F1 Score =  0.592904743237946
```

Fig. 25. Performance of the Gaussian Naive Bayes classifier



```
Accuracy =  0.824842524035805
Precision =  0.7770065075921909
Recall =  0.6257861635220126
F1 Score =  0.6932455970582543
```

Fig. 26. Performance of the Categorical Naive Bayes Classifier on the Test Data

augmentation can be used to create equal data points belonging to both classes.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, “Classification” in *An Introduction to Statistical Learning,* New York, Springer, 2013.

[2] Richard O Duda, Peter E Hart, and David G Stork, “Bayesian Decision Theory” in *Pattern Classification*, John Wiley & Sons, 2006.