Assignment 3: Predicting Potential Earnings of Adults: A Naive Bayes Classifier Approach

Boneshwar V K
Department of Biotechnology
Indian Institute of Technology, Madras
bs20b012@smail.iitm.ac.in

***Abstract*—This paper investigates the application of the Naive Bayes classifier in predicting the potential earnings of adults based on the infamous adult dataset encompassing various parameters and demographic details. The Naive Bayes algorithm is leveraged for its efficacy in handling categorical data and its ability to make probabilistic inferences. The study includes a thorough preprocessing phase for feature selection and data conditioning. Experimental results on the dataset demonstrate promising predictive performance, highlighting the suitability of the Naive Bayes classifier for such applications in socio-economic modeling and policy analysis. The findings underscore the potential of this approach in aiding decision-making processes related to workforce management and economic planning.**

***Index Terms*—categorical data, probabilistic modeling**

I. Introduction

In the realm of Data Science, one of the fundamental tasks involves categorizing the output variable, a task that varies depending on the specific problem at hand. For instance, in medical imaging, a machine learning model might discern a person's susceptibility to eye cancer through an analysis of eye images, classifying individuals as either vulnerable or not. This classification process falls into the domains of supervised and unsupervised learning, contingent upon the availability of labeled data.

Among classification models, the Naive Bayes Classifier stands out as a straightforward yet powerful tool. Leveraging Bayes' Probability theorem, this model calculates probabilities associated with output classes. While it is less commonly observed in practical applications due to its assumption of predictor independence, it serves as an excellent introduction to both classification models and probabilistic modeling.

In this study, we focus on employing the Naive Bayes Classifier to address a specific classification challenge: estimating an individual's earning potential based on demographic information. The task involves categorizing individuals into two groups: those earning more than $50,000 annually constitute one class, while the remaining individuals comprise the second class.

This paper aims to provide a foundational grasp of the Naive Bayes classifier. We commence by delving into the mathematical underpinnings of the model before applying it to tackle the stated classification problem.

II. Naive Bayes Classifier

In this section, we understand the need, potential, and mathematical background of the Naive Bayes classifier.

A. Why Naive Bayes theorem?

The Naive Bayes theorem offers distinct advantages over linear and logistic regressions in specific prediction tasks, owing to its foundational assumptions and computational effectiveness. Unlike linear and logistic regressions, Naive Bayes relies on the presumption of predictor independence, which proves highly advantageous in situations where features display minimal correlation. This assumption empowers Naive Bayes to accommodate numerous predictors without encountering the multicollinearity challenges that frequently afflict regression models. Furthermore, Naive Bayes necessitates a notably smaller number of parameter estimations, enhancing its computational efficiency and suitability for resource-constrained environments.

Additionally, Naive Bayes demonstrates exceptional efficacy in tasks like text mining and natural language processing, where term occurrences may largely stand independent. This positions it as the preferred choice for applications such as sentiment analysis and spam filtering. To sum up, Naive Bayes shines in settings characterized by predictor independence, prioritizing computational efficiency and simplicity, thus cementing its status as an invaluable tool in predictive modeling

B. Understanding of Bayes' Theorem: Mathematical background

Let us begin by looking at the mathematical equation of Bayes' probability theorem.

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

Note that $P(A|B)$ is the conditional probability. Here, $P(A)$ is called the prior of A, and $P(A|B)$ is called the posterior probability of B

We can use the same principle in our data set. Consider the following:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \tag{2}$$

Here, y represents the output variable, and X encompasses the predictor variables, i.e., $X = (x_1, x_2, \ldots, x_p)$. In this

context, $P(y|X)$ signifies the probability of y belonging to a specific class when the values of the predictor variables, X, are already known.

The Naive Bayes Classifier operates under a set of assumptions:

1) It is assumed that the predictor traits are independent.
2) Every feature is anticipated to have an equal impact on the result.

We understand that if two variables A and B are independent, then $P(A, B) = P(A)P(B)$. Leveraging this, we can re-express (2) as:

$$P(y|x_1, x_2, \ldots, x_p) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_p|y)P(y)}{P(x_1)P(x_2)\ldots P(x_p)}$$

This can be represented as:

$$P(y|x_1, x_2, \ldots, x_p) = \frac{P(y)\prod_{i=1}^{p} P(x_i|y)}{P(x_1)P(x_2)\ldots P(x_p)} \tag{3}$$

Here, the denominator being constant for a given input, we can now re-write it as:

$$P(y|x_1, x_2, \ldots, x_p) \propto P(y)\prod_{i=1}^{p} P(x_i|y) \tag{4}$$

We say the output y is now the class with maximum probability. This makes us to conclude with the following equation:

$$y = argmax_y P(y)\prod_{i=1}^{p} P(x_i|y) \tag{5}$$

The Naive Bayes Classifier's fundamentals can be summed up in this way. We now need to comprehend how we arrive at $P(x_i|y)$. Remember that $P(y)$ is just the proportion of instances of a certain class to all of the data points. The method for calculating $P(x_i|y)$ changes depending on the kind of predictor variable.

C. Obtaining Probabilities with Discrete Data

When dealing with a discrete predictor variable, determining the probability $P(x_i|y)$ becomes a straightforward process. It involves mapping the frequencies of distinct values of x_i to the corresponding output class. By leveraging these counts, we can effectively compute the probabilities associated with each value. This approach ensures a clear and systematic method for estimating probabilities in scenarios where predictor variables take on discrete values. Consider Table I.

TABLE I
FINANCIAL SELF-SUFFICIENCY OF MEN AND WOMEN

	Gender			
	Yes	**No**	**P(Yes)**	**P(No)**
Male (Men)	10	4	10/16	4/12
Female (Women)	6	8	6/16	8/12
Total	16	12	1	1

We can comprehend how $P(x_i|y)$ is calculated from Table I. The remaining predictor variables follow a similar pattern. The probability for discrete variables is modeled in this way by Naive Bayes.

D. Probabilities with Continuous Data

Creating bins is one method for dealing with continuous finite data and turning it into discrete data. We sort them into containers and then categorize them as low, high, extremely high, etc.

Assuming a distribution for the variable is a common approach to dealing with continuous data. The Normal Distribution, often referred to as Gaussian Distribution occasionally is the most typical distribution. Thus, we represent the probability as:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp(-\frac{(x_i - \mu_y)^2}{\sigma_y^2}) \tag{6}$$

This is used in the Gaussian Naive Bayes Classifier. This paper assumes Gaussian distribution for continuous variables rather than converting continuous data into discrete by creating bins.

III. THE PROBLEM

A. The Data

The dataset under consideration encompasses a diverse range of demographic attributes such as age, industry of employment, educational background, marital status, occupation, race, gender, and weekly working hours, among others. Armed with these well-defined predictors, our primary objective is to ascertain whether an individual's annual income exceeds the threshold of $50,000. This predictive task relies on leveraging the relationships and patterns observed within this comprehensive set of demographic features to make accurate income estimations.

So first, we define our output Y as the following:

$$Y = \begin{cases} 1, & \text{if the person's annual income} > \$50{,}000. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Now, the individual's projected earnings are forecasted based on established demographic attributes serving as predictor variables.

While the dataset contained a multitude of predictors, it was discerned that only a select few demonstrated interdependencies. Consequently, these correlated variables were excluded from the analysis, aligning with the Naive Bayes Classifier's underlying assumption of predictor independence. Additionally, it is crucial to acknowledge that as the number of predictor variables escalates, the model's inherent bias tends to increase, necessitating a more intricate model to effectively capture the nuances of the data. Thus, it is advisable to focus solely on the pertinent predictors for optimal model performance.

The plots are visualized and stated in the next section and that gives the insights of our data cleaning, processing, and analyzing.

The information about each of the employed predictors is compiled.

B. Impact of Gender

The data on annual income exceeding $50,000 underscores a notable impact of both gender and ethnic race counts. Among the racial categories, White individuals stand out with the highest count, totaling 27,815. Following this, we observe 3,124 Black individuals, 1,039 Asian-Pacific-Islanders, 311 Amer-Indian-Eskimos, and 271 classified under 'Other' races. It is evident that Whites and Asian-Pacific-Islanders are more prevalent in the category of earners exceeding $50,000 annually, signifying a prevailing trend. Conversely, Black individuals, alongside those categorized under 'Other' races, exhibit significantly lower representation in this higher income bracket. This disparity underscores the influence of both race and ethnicity in shaping economic outcomes, necessitating targeted strategies to address the underlying factors contributing to such discrepancies. The following table II shows the count of the population in different races.

TABLE II
ETHNIC RACE AND NUMBER OF PEOPLE

Ethnic Race	Number of People
White	27815
Black	3124
Asian-Pac-Islander	1039
Amer-Indian-Eskimo	311
Other Race	271

Moreover, when we examine the data through the lens of gender, a similar trend emerges. The population of males surpasses that of females in the category of individuals earning more than $50,000 annually. This suggests a gender-based discrepancy in income distribution, with males being more likely to fall into the higher income bracket. This pattern highlights the importance of initiatives aimed at achieving gender equality in the workplace and closing the income gap between males and females. Addressing these disparities is essential for creating a more inclusive and equitable society, where opportunities for economic prosperity are accessible to all, regardless of gender or ethnic background.

C. Data Rendering and Visualisations

Let us look at box plots between continuous variables and output to get a preliminary understanding.

Examining Figure 1, it becomes evident that individuals with a higher earning potential tend to possess elevated average ages, and educational attainment in terms of years, and dedicate more hours per week to work. These observations are gleaned from the analysis of continuous data. Now, let's delve into the associations between categorical variables and one's earning potential.

Fig. 2 shows earning potentials across various categories of people. We can observe a few interesting results like self-employed people are more likely to earn better annual income.

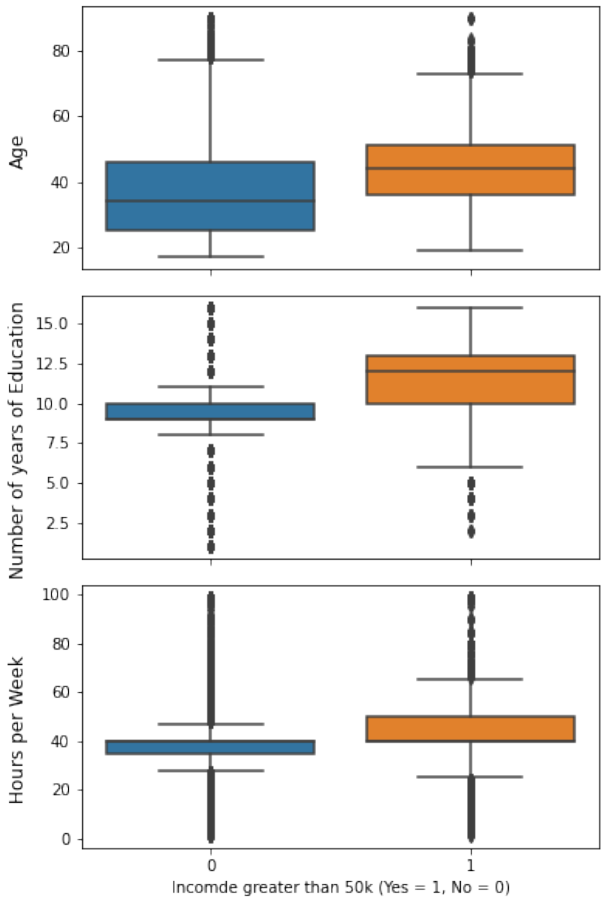

Fig. 1. Box Plots of the Data

Married people are more likely to make better incomes than unmarried or divorced people.

These visualizations help us derive some preliminary inferences. We always need to understand the data before delving deeper.

D. Utilizing Naive Bayes Estimator

Employing the Naive Bayes Estimator presents a nuanced challenge given the hybrid nature of the dataset, encompassing both discrete and continuous data variables. Calculating $P(x_i|y)$ becomes a delicate task in such a scenario.

One approach entails discretizing the continuous data through the creation of bins. A more sophisticated strategy involves training two distinct models: one exclusively on discrete or categorical data, and the other solely on continuous (numeric) data. The predicted probabilities from both models are then multiplied, adhering to the Naive Bayes Classifier's assumption of variable independence.

In this study, we focus on the latter method. This involves training a model exclusively on discrete predictors and another exclusively on continuous predictors. The combined results from both models yield the final predictions. The outcomes of the Naive Bayes Classifier are summarized in Table III.

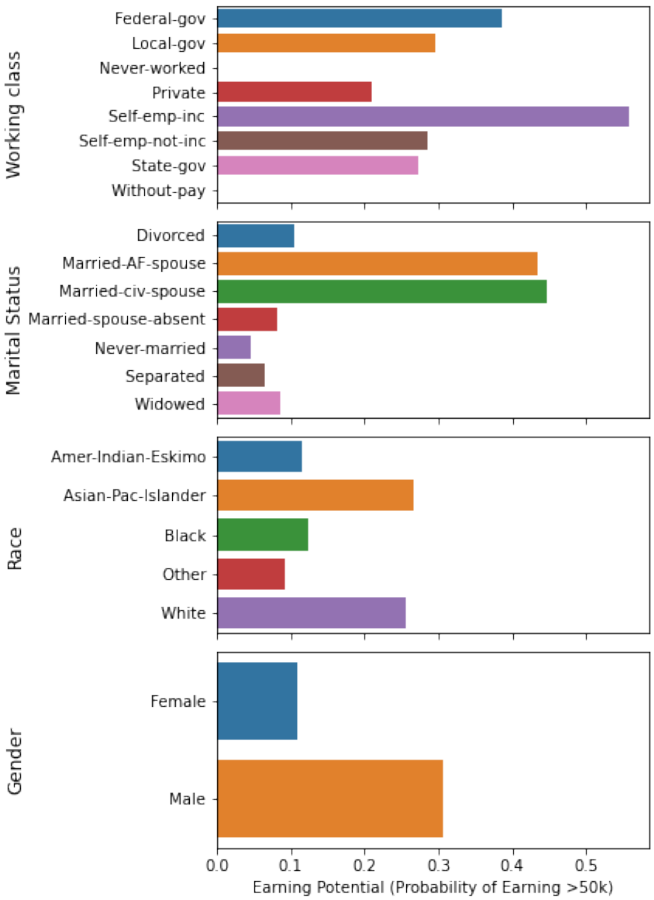

Fig. 2. Bar Plots of the Data

TABLE III
SUMMARY OF NAIVE BAYES CLASSIFIER

Model	Accuracy on Train set	Accuracy on Test set
Categorical Naive Bayes (Only Discrete Data)	79.45%	79.36%
Gaussian Naive Bayes (Only Continuous Data)	79.75%	80.83%
Combined Naive Bayes (Combining Predictions)	80.18%	80.25%

The accuracy of each model is found to be insufficient. This discrepancy arises from the underlying assumption of uncorrelated predictor variables, which, in reality, are interrelated.

Consider the correlation between age and marital status, where older individuals tend to have a higher likelihood of being married. Such real-world scenarios frequently deviate from the crucial assumption of independence, thereby affecting the model's performance..

IV. EXPLORING METHODS OF HANDLING CATEGORICAL DATA

We have seen that one way of converting continuous data into discrete or categorical data is by creating bins. This can be done whenever required.

We frequently come across cases where we have to convert categorical variables into numeric data for various reasons. This could be due to the software that we are using, for example sci-kit learn cannot deal with categorical data in string format. On the other hand the reason could be to improve model performance and how the model interprets categorical data.

A. Ordinal v/s Nominal Categorical Data

Ordinal Categorical data can be ordered, for example ranks: Students can be divided into three groups of ranks, Rank A, Rank B and Rank C, these groups could be divided based on their marks hence these form an ordinal class of categorical data.

On the other hand the data could not have any ordinal and solely for naming or labeling purpose. Example : Male, Female; Ethnicity; Marital Status,etc. This type of unordered data is called as nominal data.

When we have ordinal data, we can just encode the data with numbers of respective order. For example if Rank A has higher significance than Rank B and Rank B has higher significance than Rank C then we encode these both with 0, 1 and 2 respectively.

So, for example if we are looking at a regression model, if Rank C has tendency to give higher output then the coefficient is high and vice-versa. Hence encoding Ordinal data makes sense.

B. Encoding Nominal Data: One Hot Encoding

When it comes to Nominal data, which does not have any order we cannot simply encode it with integers. One of the most popular methods is to perform One-hot encoding, which converts a categorical feature into two different categorical features which take 0 and 1 as the only values.

TABLE IV
AN EXAMPLE OF NOMINAL CATEGORICAL DATA

Index	Gender
1	Male
2	Female

For example, consider Table IV, the variable 'Gender' consists of two values male and female. These are unordered. Hence we perform one-hot encoding.

Table V shows the data after one-hot encoding. It generates new columns 'Male' and 'Female' the values are 1 in male column and 0 in female column if the person is male and vice-versa.

One-hot encoding creates a new predictor variable. Hence the model looks at two new variables. This is necessary when the data cannot be ordered.

TABLE V
ONE HOT ENCODED NOMINAL DATA

Index	Male	Female
1	1	0
2	0	1

C. Drawbacks of One-hot Encoding

This paper deals with a problem which has nominal categorical data. We did not use one-hot encoding to predict the outcome. The reason for this is understood from the mathematics behind Bayesian classifier, it needs the categorical variable to compute probabilities. Creating two new variables will turn discrete data into continuous and we end up using Gaussian estimates for probability.

Hence sometimes One-hot encoding is not necessary at all. One of the drawbacks of one-hot encoding is the increase in data dimensions. Consider a categorical variable with 100 classes, this adds 100 new variables to the data.

So, One-hot encoding is not always preferred. Some alternatives include CatBoost encoder which is similar to a nominal encoder except the integers allocated are obtained by computing weights through the target variable and higher level encoding methods like hash encoding, effect encoding and binary encoding.

V. CONCLUSION

We understood the simplest classification model. We use a fundamental probability theorem to predict the output class. While the model is not used in real-life scenarios, the Naive Bayes Classifier is an excellent starting point to understand the working of its complex versions like Linear Discriminant Analysis or Logistic regression.

Naive Bayes Classifier gives us a good understanding that probability and linear algebra play a fundamental role in machine learning models. We often tend to overlook the fundamentals when we look at complex models. We need to remember that these models were built upon fundamental theorems and principles like Bayes' probability theorem. We can also come across probability when we look at optimization tasks. One famous example is the maximum likelihood estimate which is used in logistic regression.

Hence, we need to understand probability and linear algebra before we delve deeper into any of the machine learning algorithms.

REFERENCES

[1] Chris Albon. Machine Learning with Python Cookbook. O'Reilly Media, Inc, 2018.

[2] Geek for Geeks, 2017. Naive Bayes Classifier, Geek for Geeks. Accessed on Oct. 15, 2021. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[3] Analytics Vidhya, 2015. Naive-Bayes for mixed typed data in scikit-learn - Analytics Vidhya. Accessed on Oct. 15, 2021. [Online]. Available: https://medium.com/analytics-vidhya/naive-bayes-for-mixed-typed-data-in-scikit-learn-fb6843e241f0