Naive Bayes' Classifier

Devcharan K
Indian Institute of Technology, Madras

***Abstract*—Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. In this paper, data about the adult income taken from the 1994 Census bureau database by Ronny Kohavi and Barry Becker was analyzed using Machine Learning techniques such as Logistic Regression and Naive Bayes Classifier. The performance of both these machine learning algorithms is tested and compared.**

I. Introduction

Jane Austen once said, "A large income is the best recipe for happiness I ever heard of." Income is instrumental in deciding a person's standard of living and the financial status in the society. It plays a key role in determining the growth of a nation. The aim is to identify meaningful insights from the given data to predict if a given individual earns more than $50K per year or not.

Naive Bayes classification is a straightforward and a powerful algorithm for the classification task. Naive Bayes classification is based on applying Bayes' theorem with strong independence assumption between the features. Naive Bayes classification produces good results when we use it for textual data analysis such as Natural Language Processing. Naive 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. Naive Bayes classifier applies Bayes' theorem in practice.

In this paper the data about the US adult population such as Age, Education, Occupation, Weekly working hours etc. is considered. The aim is to classify the data based on the characteristics into a group of individuals that earn more than $50K per year and a group of individuals that earn less than $50K per year. A Naive Bayes classifier algorithm will be used as well as Logistic Regression to create machine learning algorithms to predict the group a person falls into based on the given data.

In Section 2, Naive Bayes' Classifier and its applications are explained. In Section 3, the problem and the method to solve it is shown. Section 4 is an Exploratory analysis of the cleaned data-set with the usage of Python packages namely Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn and visualizations of the same. In Section 5, the concepts used to prepare the model is evaluated. Finally the inferences of the analysis is presented in Section 6.

II. Naive Bayes' Classifier

Naive Bayes' Classifier is a form of Supervised Learning. Naive Bayes Classifier uses the Bayes' theorem to predict membership probabilities for each class such as the probability that a 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)**.

Bayes' theorem is given by:

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

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

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

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. Naive 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.

There are three types of Naive Bayes' Classifier algorithms:

A. Gaussian Naive Bayes'

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian 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 μ_i be the mean of the values and let σ_i be the variance of the values associated with the i^{th} class. Suppose we have some observation value x_i . Then, the probability distribution of x_i given a class C_k can be computed by the following equation –

$$p(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}} \tag{3}$$

Sometimes the distribution of class-conditional marginal densities need not be normal. In these cases, the other methods must be used to determine the probability distribution of the dataset before proceeding.

B. Multinomial Naive Bayes'

With a Multinomial Naive Bayes model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial $(p_1, ..., p_n)$ where pi is the probability that event i occurs. Multinomial Naive 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. The likelihood of observing a histogram x is given by

$$p(x|C_k) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} p_{ki}^{x_i} \tag{4}$$

C. Bernoulli Naive Bayes'

In 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.

Applications of Naive Bayes': Naive Bayes is one of the most straightforward and fast classification algorithms. It is very well suited for large volumes of data. It is successfully used in various applications such as :

- Spam filtering
- Text classification
- Sentiment analysis
- Recommender systems

III. The Problem

Census Data about adults such as Age, Education, Workclass, Occupation, Sex, Race, Marital Status and Working hours per week is available in the dataset. The aim of this exercise is to classify the people into two groups, one with people who earn more than $50K a year and another with people who earn less than $50K annually. This information can also be used to predict the income level of other samples with the help of appropriate data.

A. Data Cleaning

The analysis begins by cleaning the data by removing any missing values or any incompatible values. In this dataset the missing values are in the form of a string ' ?' in the columns Occupation, Workclass and Native Country. These question marks can be replaced with the most frequent value in the given column. There are 1836 instances where both workclass and occupation values are missing which is intuitive because if occupation is unknown, the workclass is bound to be unknown. The United States is the 'Native country' for 81% of the samples, so the missing values in 'Native country' are replaced with the USA. The data is then converted into categorical variables from continuous variables to make the analysis more straightforward.

B. Outliers

Outliers are present in almost all datasets. Outliers here are removed after analysing them by three methods namely Standard Deviation, Z_score and Inter-quartile range.

- Age : z_score or Std Deviation
- fnlwgt : z_score or Std Deviation
- Hours per week : z_score or Std Deviation
- Capital Gain : Average of different groups
- Capital Loss : Average of different groups

IV. Exploratory analysis

The target variable to be predicted is the 'Income' variable. The remaining variables are converted to categorical variables.

A. Workclass

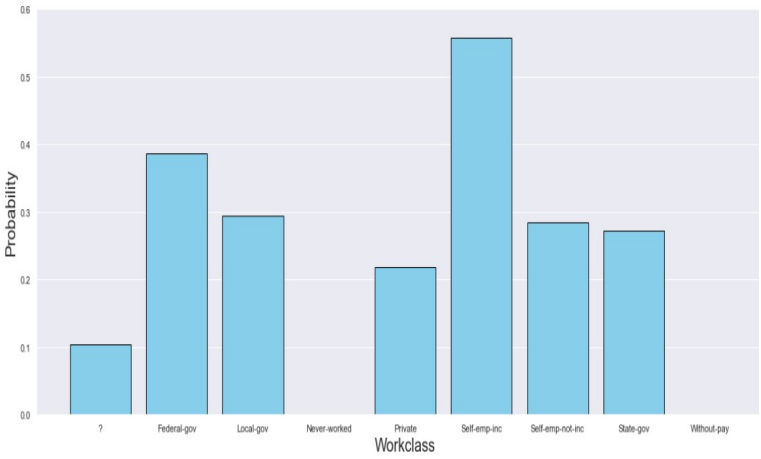

Fig. 1. Probability Distribution of having an income greater than $50K for different work classes

Fig. 1 shows that Self Employed people and government employees are more likely to earn greater than $50K annually.

B. Occupation

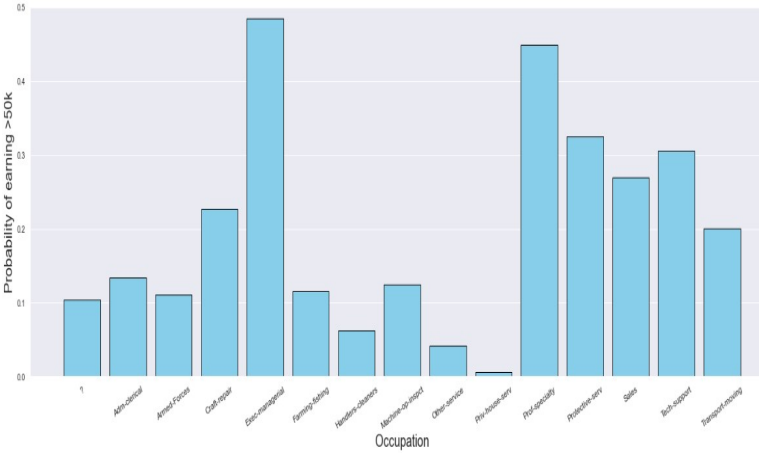

Fig. 2. Probability Distribution of having an income greater than $50K for different occupations

Fig. 2 shows that people who are Executive-Managerial roles and Prof-speciality roles are most likely to earn greater than $50K annually.

C. Marital Status and Relationship

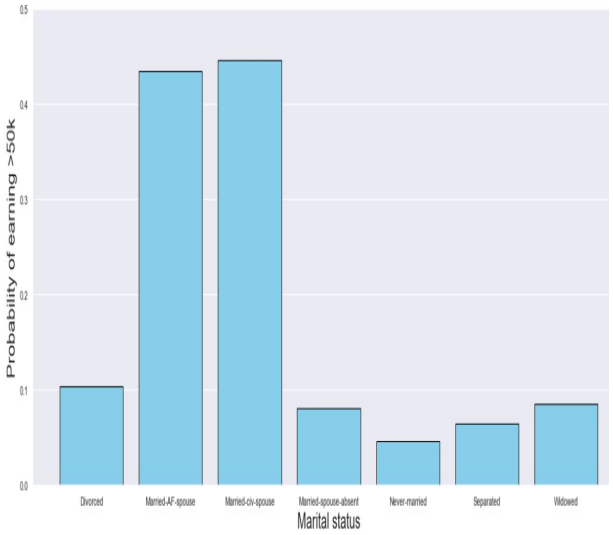

Fig. 3. Probability Distribution of having an income greater than $50K with respect to marital status

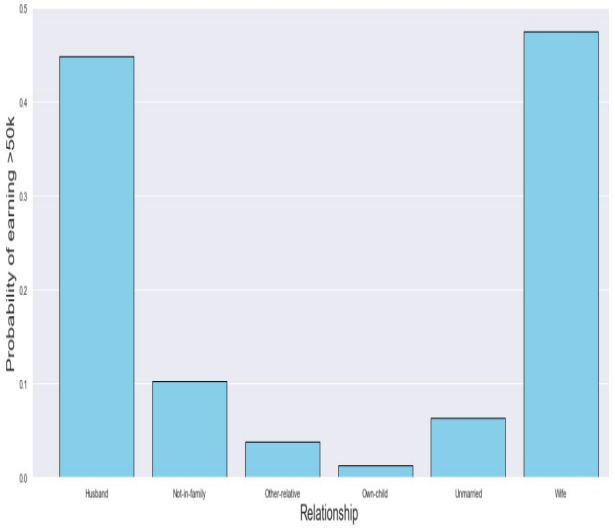

Fig. 4. Probability Distribution of having an income greater than $50K with respect to relationship

Fig.3 and Fig. 4 show that people who are married have a higher chance of having above $ 50K income.

D. Education

'Education' and 'Edu years' columns provided the exact same information as each other so 'Edu years' column was omitted.

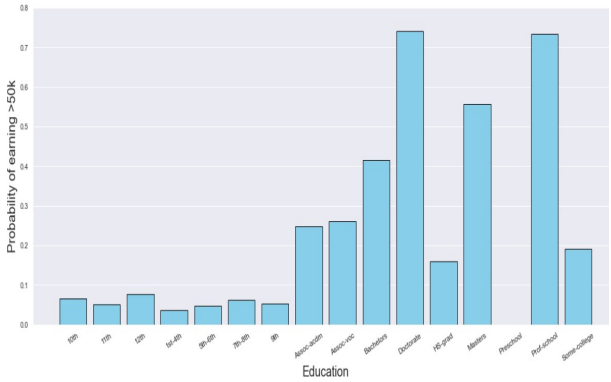

Fig. 5. Probability Distribution of having an income greater than $50K with respect to education

Fig. 5 shows that people who have done a doctorate or been to Prof school are more likely to earn greater than $50K annually followed by people with masters and bachelors degrees.

E. Race

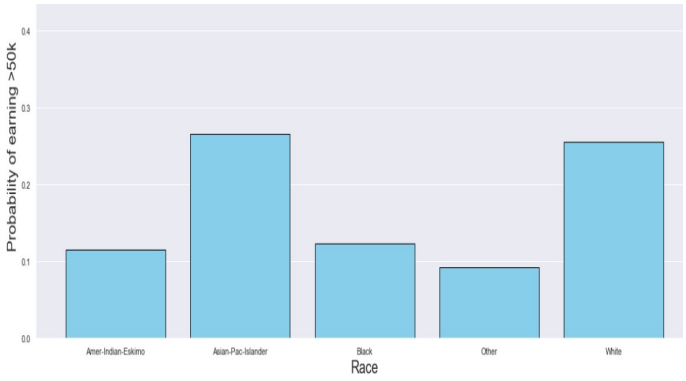

Fig. 6. Probability Distribution of having an income greater than $50K with respect to race

Fig. 6 shows that white and Asian-pac people are more likely to earn greater than $50K annually.

F. Sex

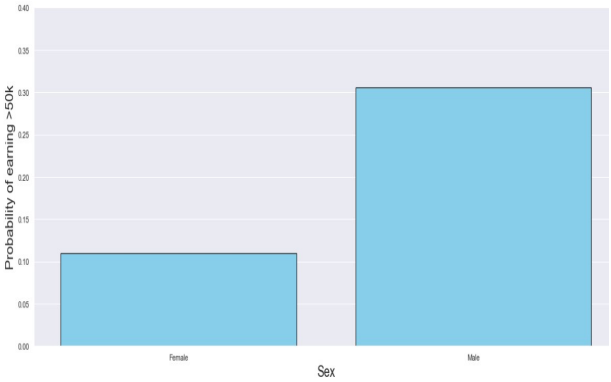

Fig. 7. Probability Distribution of having an income greater than $50K with respect to sex

Fig. 7 shows that men are more likely to earn greater than $50K annually.

G. Age

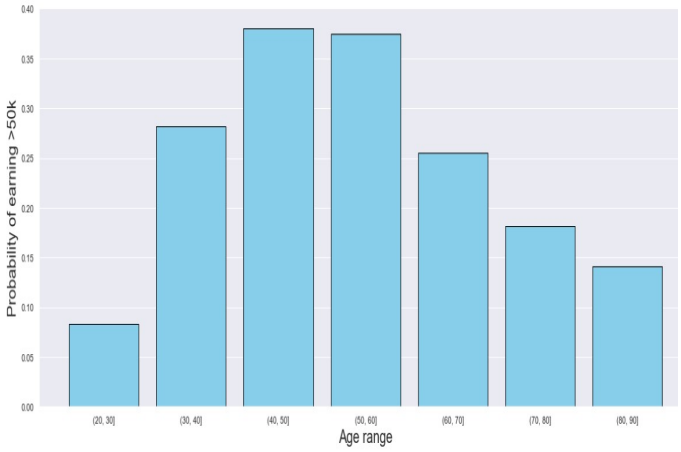

Fig. 8. Probability Distribution of having an income greater than $50K with respect to age

Fig. 8 shows that people in their 40s to 50s have the highest probability of earning greater than $50K annually.

H. Working hours

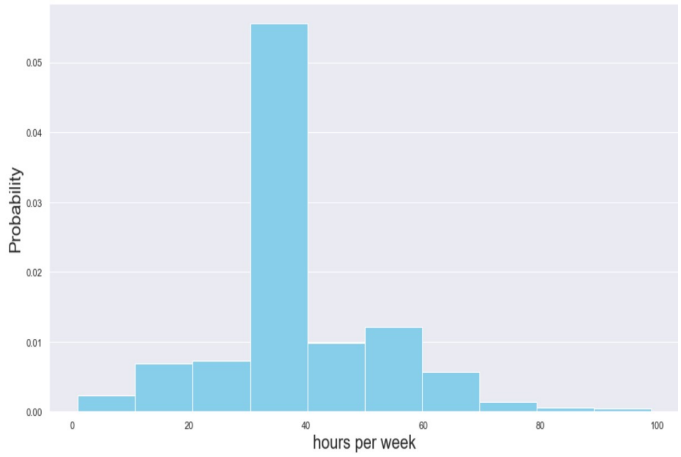

Fig. 9. Probability distribution of having an income greater than $50K with respect to number of hours of work per week

Fig. 9 shows that people working around 30 to 40 hrs a week have the most likelihood of earning greater than $50K annually.

I. Correlations

Looking at the correlation levels between all the features is an important part of classification problems because some features are more correlated than others and should be given more importance during feature selection for the model.

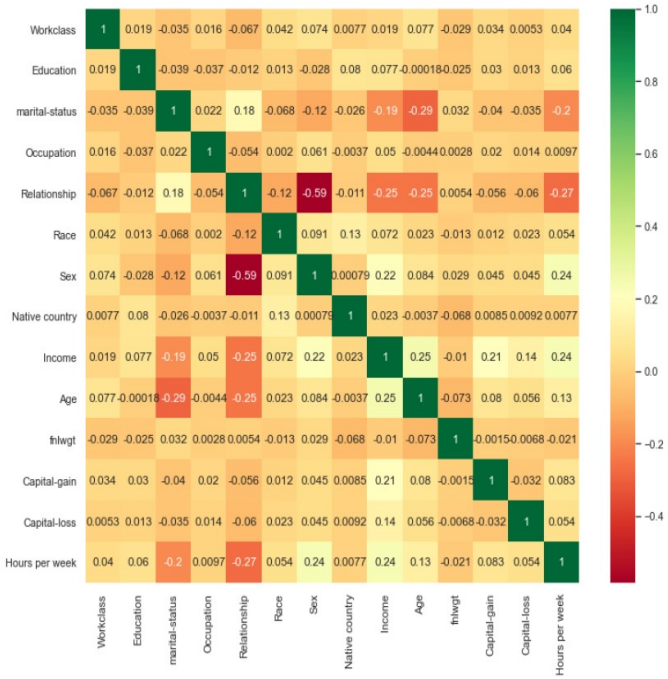

Fig. 10. Correlation heatmap of all important features in the dataset

Fig. 10 shows the correlations between all important features of the given dataset. Some important correlations are given below:

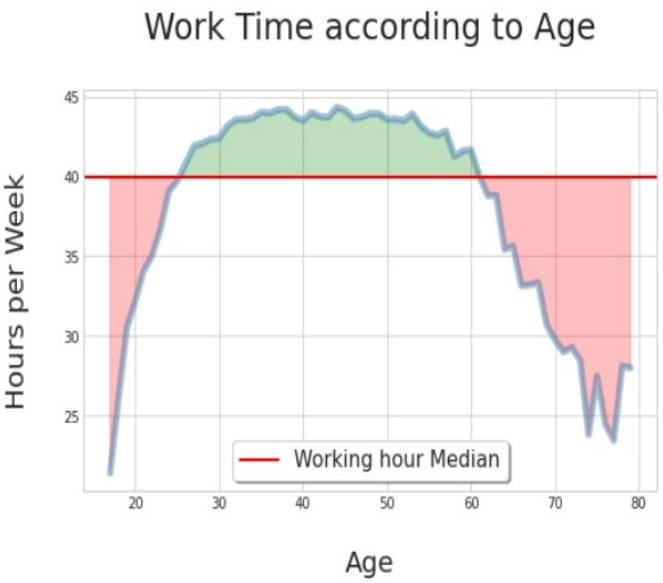

Fig. 11. Relation between Working hours per week and age

Fig. 11 shows that people in the age group 30-50 generally work longer hours than the median value

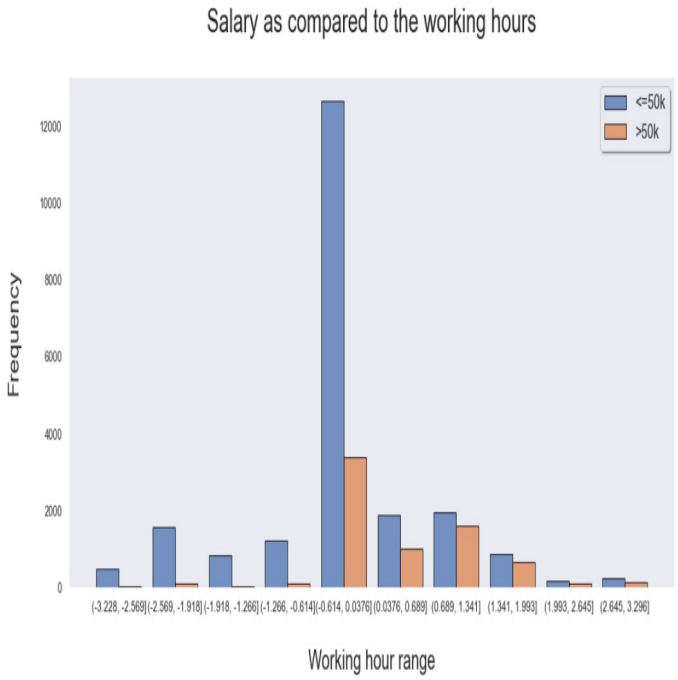

Fig. 12. Relation between Working hours per week and Income

Fig. 11 and Fig. 12 show that people who work 30 to 50 hrs a week have the most likelihood of earning greater than $50K annually.

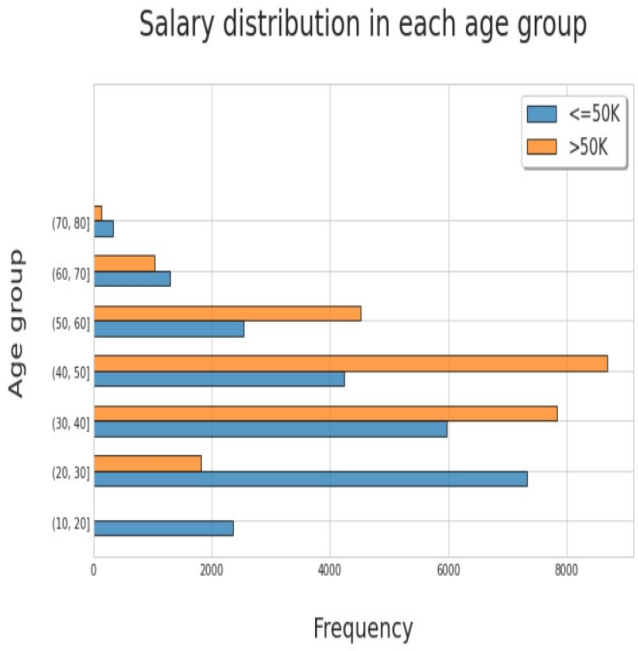

Fig. 13. Relation between age and Income

Fig. 13 shows that people in their 40s and 50s are most likely to have an income of over $ 50K.

V. Model Evaluation

A. Train Test Split

To train the machine learning models, the given dataset is split into a train dataset and a test dataset. This is done so that the model can use the data to learn and compare its prediction with the real value of the event the model is trying to predict. This procedure is necessary before applying the model on a new testing dataset to actually predict an event.

B. Logistic Regression

Logistic Regression is a Supervised machine learning algorithm that is used when classification problems have only two possible classes (like True/False). The considered dataset is of the same type, where the two possible classes are >$50K or <=$50K.

Hyperparameters are the ones that control the underfitting and overfitting of the model. Optimal hyperparameters often differ for different datasets. To get the ideal set of hyperparameters, the model is tested for each hyperparameter setting and the ones that give the best results are selected. The Confusion Matrix for the Logistic Regression model on the testing dataset before parameter tuning is: $\begin{bmatrix} 5196 & 308 \\ 1260 & 560 \end{bmatrix}$

and after parameter tuning it is: $\begin{bmatrix} 5215 & 289 \\ 1270 & 550 \end{bmatrix}$

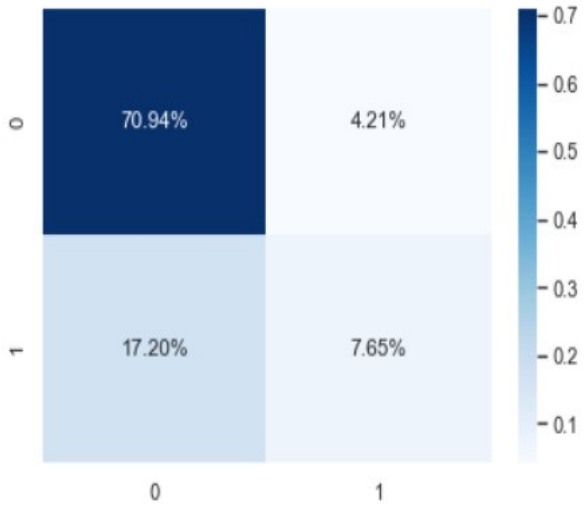

Fig. 14. Confusion matrix for Logistic Regression model

The Model accuracy is 77.82% before parameter tuning and 78.72 % after parameter tuning.

C. Gaussian Naive Bayes'

Gaussian Naive Bayes' is a classification algorithm that treats the probability distribution of the dataset as a gaussian (normal) distribution. Like the Logistic Regression case, Hyperparameter tuning is required to get the best results in Gaussian Naive Bayes' classifier.

The confusion matrix for the Gaussian Naive Bayes' model on the testing dataset before and after parameter tuning is $\begin{bmatrix} 5169 & 335 \\ 1276 & 544 \end{bmatrix}$

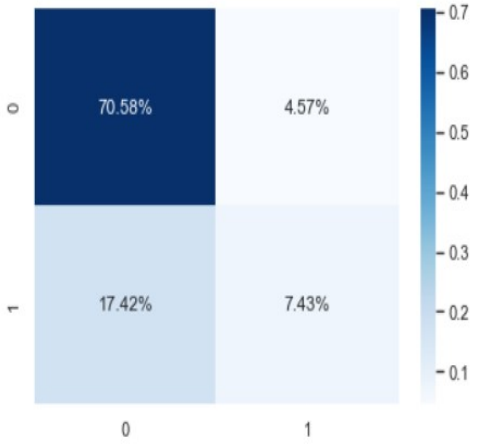

Fig. 15. Confusion matrix for Gaussian Naive Bayes' model

The model accuracy is 78%.

Confusion matrix is used to show the performance of the algorithm. The accuracy of the model can be predicted using the confusion matrix.

VI. Inferences

As shown in Fig. 10, age group ,marital status and working hours per week are the biggest factors influencing the income of any given person. People in their 30s to 50s, people who work around 30-50 hours per week and people that are married are the most likely to have an annual income of over $50K. There is considerable influence from attributes like race, sex and education on income status of the people too. In addition to the above, people with a positive capital gain have a higher probability of having an income above $50K.

Conclusion

Logistic Regression is a widely used machine learning technique especially in binary classification problems. Gaussian Naive Bayes' classifier is one of the algorithms in the family of Naive Bayes' classifiers that use the Bayes' theorem to predict probabilities for each class in the dataset. The accuracy of both these models is similar for the considered dataset on adult income which is 79% accuracy.

References

[1] "Naive Bayes' Classifier" Wikipedia [Accessed on 21 October] Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[2] "Logistic Regression" Wikipedia [Accessed on 21 October] Available: https://en.wikipedia.org/wiki/Logistic_regression

[3] Naive Bayes Classifier From Scratch in Python Available: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

[4] Evaluating Machine Learning Models using Hyperparameter Tuning Available: https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/

[5] Naive Bayes Classifier — How to Successfully Use It in Python? Available: https://towardsdatascience.com/naive-bayes-classifier-how-to-successfully-use-it-in-python-ecf76a995069

[6] Naive Bayes' Scikit-learn documentation Available: https://scikit-learn.org/stable/modules/naive_bayes.html

[7] Gaussian Naive Bayes with Hyperparameter Tuning Available: https://www.analyticsvidhya.com/blog/2021/01/gaussian-naive-bayes-with-hyperpameter-tuning/