Assignment 3
A mathematical essay on Naive Bayes classifier.

Anik Bhowmick
Inter Disciplinary Dual Degree Data-Science
Indian Institute of Technology Madras
ae20b102@smail.iitm.ac.in

***Abstract*—This assignment discusses the application of a Naive Bayes classifier on the Adult dataset. This data was extracted from the 1994 Census Bureau database by Ronny Kohavi and Barry Becker. This data indicates whether a person has an annual income of more or less than 50,000 US dollars based on various attributes like age, gender, education qualification, working class, work hours, and native country. By studying the data and fitting the model, our task is to unfold which kinds of people are in higher income groups and which are less. For that, we will adopt various visualization techniques. Also, because this data set is full of categorical columns, we will have better exposure to how to handle them.**

***Index Terms*—Visualization, K Modes, Naive Bayes classifier, One hot encoding, Confusion Matrix, Accuracy, Precision, Recall, $\mathbf{F_1}$ score, ROC.**

I. Introduction

- This assignment is a simple binary classifier about the annual salary on the US census adult data using the Naive Bayes classifier. Our main task is to unfold the underlying patterns in the data that decide people's annual income. Further, we will try to understand how income affects people's social lives, such as marital status and relationships. This data not only opens up a great opportunity to work with categorical features but also will help to understand the workings of the Naive Bayes classifier.
- Naive Bayes is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. It is a statistical algorithm that analyzes the relationship between a set of independent variables and the dependent binary variables. It is a powerful tool for decision-making. As the name suggests, 'Bayes', its working principle is based on the Bayes theorem. The name 'Naive' is because it makes a naive assumption that input features are linearly independent and each makes an equal contribution to the output, which may not be the case always.
- The main task in this assignment is to build a classifier model to find the relation between the input features and the target: whether a person makes 50,000 USD annually. To do so, we will demonstrate various data visualization and preprocessing techniques.
- This paper will demonstrate the data analysis technique relevant to the Naive Bayes classifier with the help of visual plots and mathematical equations. We will discuss various data handling processes such as categorical features, missing values, etc.

II. Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic machine learning algorithm for classification tasks. It is based on Bayes' theorem, a fundamental theorem in probability theory.

- The Working basis of Naive Bayes is similar to the maximum a posterior (MAP) estimation.
- We calculate the probability of each class as prior probability. Then, we compute the conditional probability of a feature given a class.
- Using these precomputed data, when a new data point arrives with its features, we calculate the conditional class probabilities given those features for all classes. The data point is considered to belong to that class whose class probability is the highest. These things will get clearer in the detailed mathematics part discussed below.

A. Assumptions

- It assumes the features are conditionally independent based on classes.
- Each feature has an equal contribution to the class label.

Because the Naive Bayes classifier uses probability, it can work on categorical and numerical data because the frequency of a particular instance matters in probability.

B. Bayes Theorem

Given two events, A and B, if we know the conditional probability $P(A|B)$, Bayes theorem will enable us to calculate the inverse probability as :

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

In the context of Bayesian estimation and Naive Bayes:

- $P(B)$ is called prior probability.
- $P(A|B)$ is called Likelihood probability.
- $P(B|A)$ is called posterior probability.
- $P(A)$ can be treated as a normalizing constant.

C. Detailed Mathematical analysis

With regards to datapoint, consider $X = (x_1, x_2, ...x_n)$ and we want to find its label y. n is the number of features. Let's denote the class probability (prior) by $P(y)$. $P(y)$ we can find from the training data class labels. For a particular class, it can be computed as :

$$P(y = \text{class}_i) = \frac{\text{Number of instances of class i}}{\text{Total number of instances}}$$

For each feature conditional probability of each instance, we can get a similar way. For the new data point, we will use the Bayes theorem:

$$P(y|x_1, x_2, ...x_n) \propto P(y)P(x_1, x_2, ..., x_n|y)$$

$$P(y|x_1, x_2, ...x_n) \propto P(y) \prod_{i=1}^{i=n} P(x_i|y)$$

The above equation is due to the independence of features. The class is chosen which has the highest probability. Unlike other models, it does not have an explicit training and cost function optimization process. This makes the model training quite faster.

D. Evaluation Metrics

There are several evaluation metrics available to evaluate a logistic regression model.

- **Confusion Matrix** A confusion matrix is the table often used to describe the performance of a classification model on a set of test data for which the true values are known. A confusion matrix looks exactly as given below.

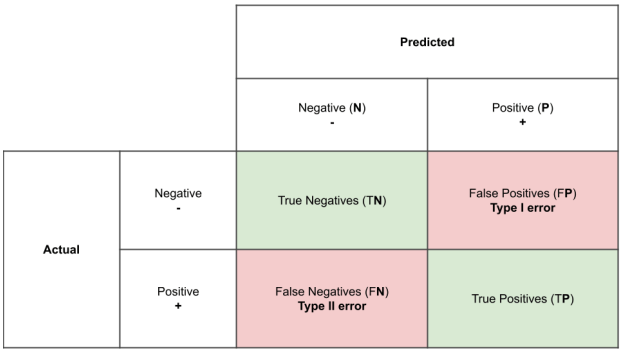

Fig. 1. Confusion matrix

TP: True positive is how many positives are true as predicted by the model.
FP: False positive is how many are predicted to be positive but not positive.
FN: False negative is how many are falsely predicted to be negative.
TN: True negative is how many are predicted to be negative, which are actually negative.
It is clear that only TP and TN are the correct predictions made by the model. The rest are wrong predictions. A good model should have FP and FN as small as possible.

- **Accuracy** We define accuracy as the fraction of correct prediction out of the total prediction given by the model.

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

A good model should have high accuracy. But that is not a necessary condition. Even a highly accurate model can give more wrong predictions if trained on a highly imbalanced dataset (i.e., the number of one class instances is much larger than the other). So, accuracy alone can not help us decide whether a model is good or not.
- **Precision**: This is defined as how many are actually positive out of total positive prediction.

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

- **Recall**: or true positive rate is defined as the number of predicted positive out of actual positive. It is also known as Sensitivity.

$$\text{Recall (TPR)} = \frac{TP}{TP + FN}$$

Increasing recall decreases precision, often known as "precision-recall trade-off". It depends from problem to problem what we want: more precision or more recall. Like in the medical field of cancer disease detection models, we want the model to become robust to detect cancer. In this case, predicting a negative, even if a patient is positive, is highly undesirable. So, we want a more positive rate, meaning high recall. But in the present data, we need not look into the details of these, instead we can just check the respective F_1 scores.
- $\boldsymbol{F_1}$ **score** : Is the harmonic mean of precision and recall.

$$\boldsymbol{F_1}\ \textbf{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

If we don't want to emphasise the precision and recall individually, we check for a high F_1 score for a good model.

III. The Data

Brief information about the data:

- Total data points are 32561. Columns are 'Age', 'Workclass', 'fnlwgt', 'Education', 'Education_num', 'Marital_status', 'Occupation', 'Relationship', 'Race', 'Sex', 'Capital_gain', 'Capital_loss', 'Hours_per_week', 'Native_country', 'Income'. Please note the original data didn't have any column names. So, we had to add these names on our own based on the information available.
- Some of the categorical columns are 'Workclass', 'Education', 'Marital_status', 'Occupation', 'Relationship', 'Race', 'Sex', 'Native_country', and 'Income'. So, almost the majority are categorical
- Many of these categorical columns have missing entries in the form of '?.' marks.

Variable	Definition	Key
age	Age	Continuous
workclass	workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
fnlwgt		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	No. of years of education	Continuous
marital status	Marital status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Occupation	occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Farming-fishing, Adam-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
relationship	Relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
race	Race	White, Asian-Pac-Islander, Amer-Indian-Eskmo, Other, Black
sex	Gender	Female, Male
capital-gain	Capital gain	Continuous
capital-loss	Capital loss	Continuous
hours-per-week	Working hours / week	Continuous
native-country	Native Country	United-States, Cambodia, England. . .

TABLE I
DESCRIPTION OF THE DATA

IV. THE PROBLEM

As mentioned, our goal in this assignment is to predict whether a person earns above 50k dollars by Naive Bayes. We followed the following steps to prepare the data before feeding it into the model.

A. Cleaning and preparing the data

The dataset has almost entirely categorical data in the form of strings.

1) Handling Missing values: After thoroughly studying, we found that the categorical columns containing missing values are: 'Workclass', 'Occupation', and 'Native_country'. They have '?' in the place of missing values. We used an unsupervised algorithm K-prototypes to impute these missing values. It is a clustering technique that works for both numerical and categorical columns. For numerical missing values, we can assign respective cluster mean, and for categorical missing values, we can use cluster modes. We won't discuss the details of this process in this assignment. Below is a brief description of our missing value imputation technique.

- First, drop all the columns which have missing values. Now, with the remaining data, fit the k prototype with the suitable number of clusters.
- Once the clusters are formed, check for each missing column data to which cluster it belongs. Once that is found, use the mode of that particular cluster for that particular column to replace the missing entry.

In my case, I experimented with several clusters. I found 4 clusters gave better model results. So, I chose 4 clusters.

2) Encoding the categorical columns: Because any ML model can't work with strings, we need numerical representations of the categorical features. For this purpose, we decided to use one hot encoding. One hot encoding converts the categorical data to vectors containing zeros and ones. For example, when we apply one hot encoding to the gender column, it will make vectors (1,0) and (0,1). The first vector may denote male, and the second one will be female or vice versa. Here, we got two components of each vector because the gender column has only two kinds of instances: male and female. If any column has n different instances, we will have n different one-hot vectors, each having n components.

B. Exploratory analysis

1) Correlation among the numerical featurers: Two features are correlated if they have an absolute correlation coefficient close to 1. They are uncorrelated if their correlation coefficient is closer to 0. If a model is fed with correlated data, model training will be unstable, and overfitting may become predominant. So, the usual technique to get rid of correlation is to drop the highly correlated features. The formula for correlation between two features x_i and y_i is :

$$r_{xy} = \frac{\sum_i x_i y_i - n\bar{x}\bar{y}}{\sqrt{\sum_i x_i^2 - n\bar{x}^2}\ \sqrt{\sum_i y_i^2 - n\bar{y}^2}}.$$

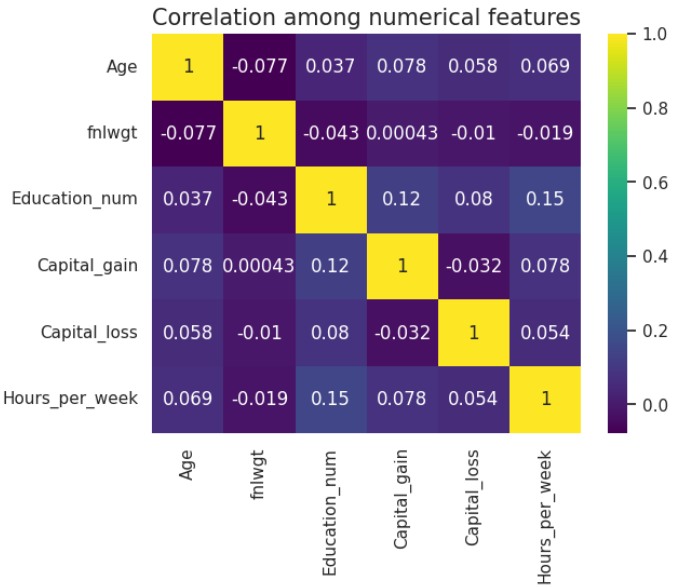

Fig. 2. Correlation plot

But in our case, all the numerical features are almost uncorrelated. Because the values are quite close to zero, this is really helpful for us, as we don't need to drop any numerical features. The Naive Bayes classifier will work better on such data because it considers all the features equally important. In the case of categorical features, because we will encode them by one hot encoding technique, they will be uncorrelated.

2) Distribution Plots: It's sometimes useful to check the distribution of numerical features. As distribution sometimes gives very useful insights.

Fig. 3. Distribution of age

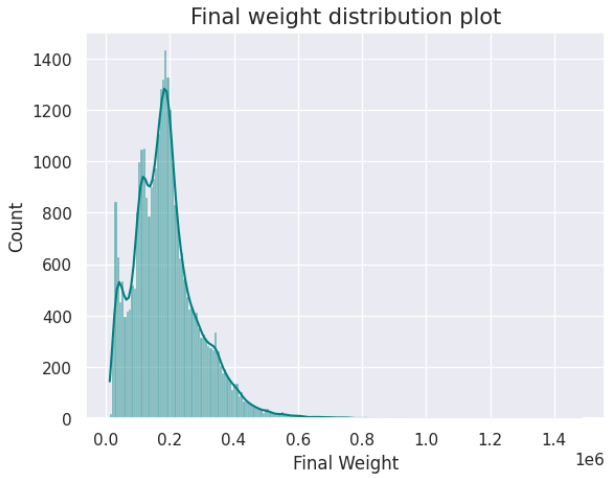

Fig. 4. Distribution of Final Weight (fnlwgt)

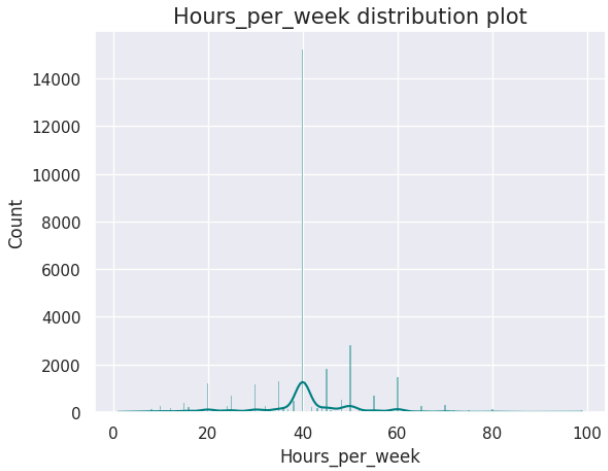

Fig. 5. Distribution of hours per week

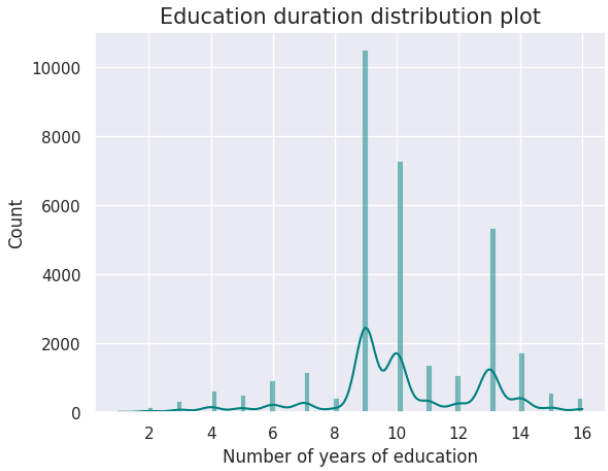

Fig. 6. Distribution of duration of education (years)

From these distribution plots, we see except for the age and final weight columns, all other columns have an uneven distribution of data (Some have very high occurrence, some are very low). This may lead to erroneous model training, so we decided to treat numerical columns as categorical columns. In this case, the model won't prioritize major data points compared to minor ones.

3) Preliminary study on the trend followed by data: We will study a few plots before finally feeding the data to the model.

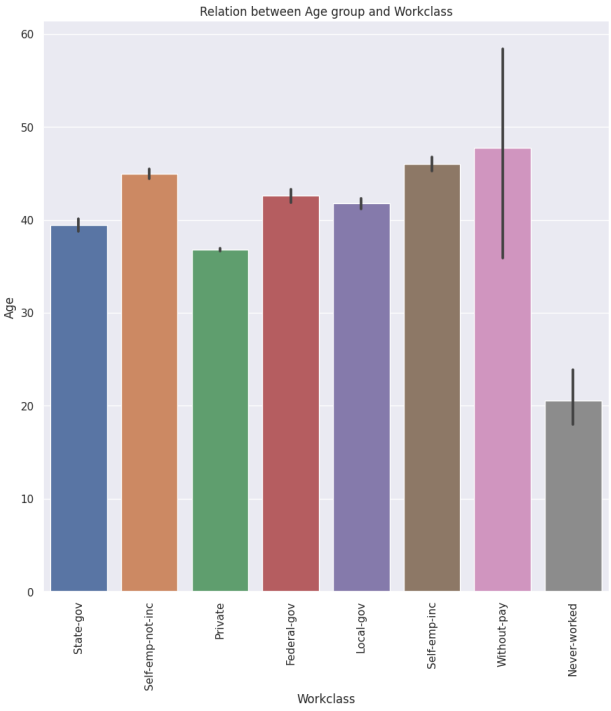

Fig. 7. Works class vs Age

From this plot, we see younger people under 20 never worked. It is as per expectation.

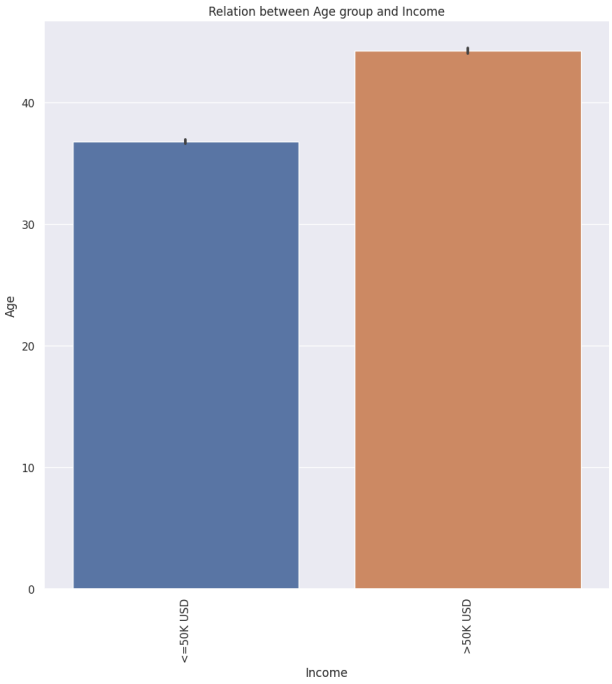

Fig. 8. Age and Income relation

All the age groups earning above 50K USD are above 40 years old. This may be because, with increasing age, working experience and exposure increase. So salary can also be expected to increase.

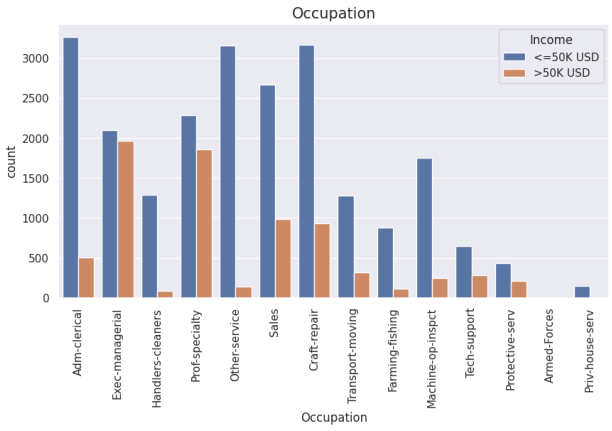

Fig. 9. Occupation count

We see people from executive and managerial occupations are more likely to earn above 50K USD. Whereas clerics, craft repairers, and other job workers are lower-income groups. Our common knowledge also tells us that executives make more money than others.

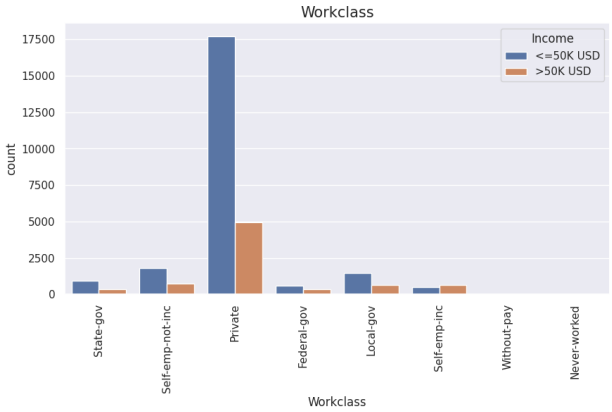

Fig. 10. Work place count

This plot reveals that people in private firms are more likely to earn above 50K USD.
The sole purpose of this kind of preliminary study is to check whether the model also follows this kind of trend after training on the validation set.

4) Visualizing the class distribution: We used a bar plot to get an idea of the number of data points greater and less than 50,000 dollars.

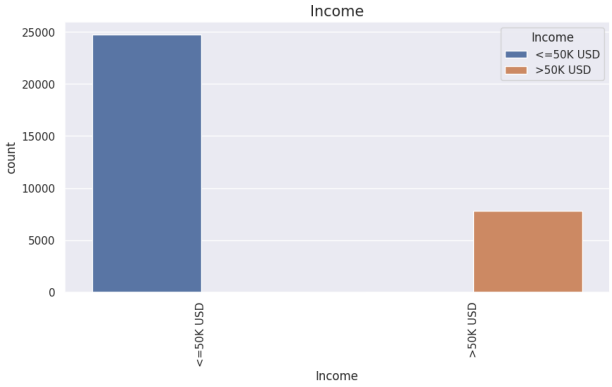

Fig. 11. Income count plot

There are 24720 people with less than 50K dollars in income and 7841 people with greater than 50K dollars. Undoubtedly, the data is highly imbalanced. During the train test split, we have to maintain this ratio of the number of data points of each class. For this purpose, we have to stratify the splitting strategy. Techniques like under-sampling and oversampling also might help. But we will keep our task simpler and easily understandable so we won't adopt those techniques.

C. Splitting the Training data to Train and validation sets

To validate the model, we decided to split the model into training and validation sets. Because the number of data points is fairly large, a 75:25 splitting ratio will work fine. This leaves us 24420 data points in the train and 8141 data points in the test set

D. Statistical model

Our model is a Complement Naive Bayes classifier from sklearn. This model works well on both categorical and numerical data. Even when the classes are highly imbalanced, the performance of this model is far better than conventional Naive Bayes classifiers such as Gaussian and Multinomial Naive Bayes. The first training round gave fairly good accuracy on the train and test set. Train set accuracy came around 87%, and for the test set, 83%.

E. Visualization and validation

Below are the evaluation metrics for the validation set:

Income	Precision	Recall	F1-score
<=50K USD	0.93	0.85	0.89
>50K USD	0.62	0.80	0.70

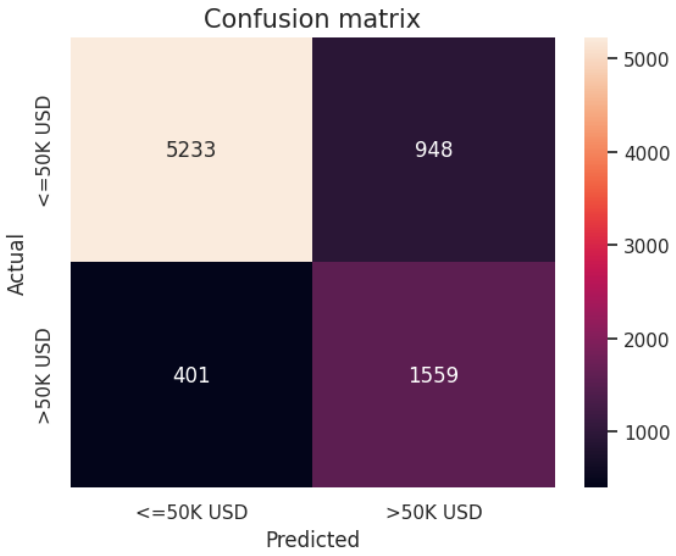

Fig. 12. Confusion Matrix

From this figure, we can infer the following

- **True Positive**=1559 (Actual above 50K USD, predicted above 50K USD)
- **False Positive**=948 (Actual below 50K USD, predicted above 50K USD)
- **False Negetive**=401 (Actual above 50K USD, predicted below 50K USD)
- **True Negative**=5233 (Actual below 50K USD, predicted below 50K USD)

From this, we get the accuracy as:

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

$$\text{Accuracy} = \frac{1559 + 5233}{1559 + 5223 + 948 + 401} = 0.83$$

The goodness of a model can be tested by one more metric called Area under the ROC (Receiver operating characteristics) curve. The more the area, the better the model is. In our case, we achieved an AU-ROC of 0.91. Which indicates the model is performing quite well.

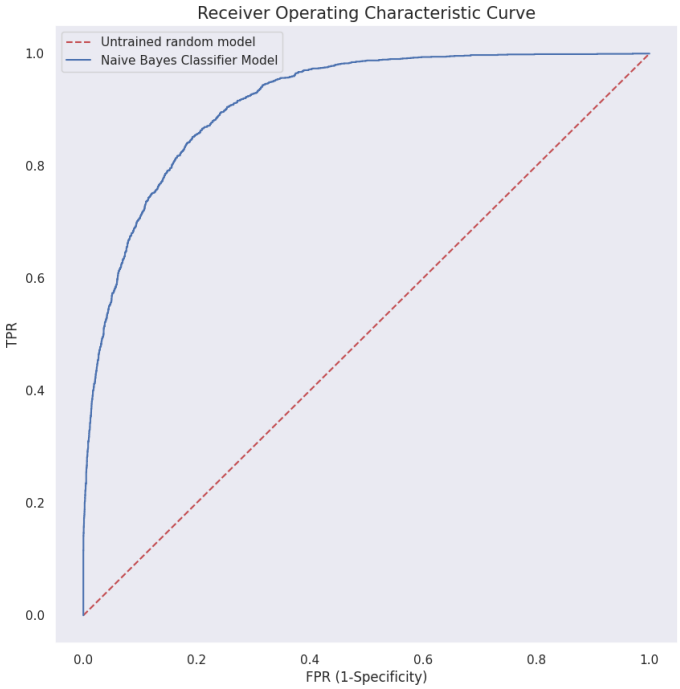

Fig. 13. Receiver operating characteristic curve

F. Insights captured by studying the data

After going through the training data, we captured some patterns. They are discussed below.

1) Insights in training data:

- From the education plot, we see that the largest income group above 50K USD is Bachelors. In the case of master's degree holders, professors and doctorates, more people earn above 50K USD than below 50K USD. The lowest income groups are school pass-outs. So evidently, education qualification matters in deciding income.

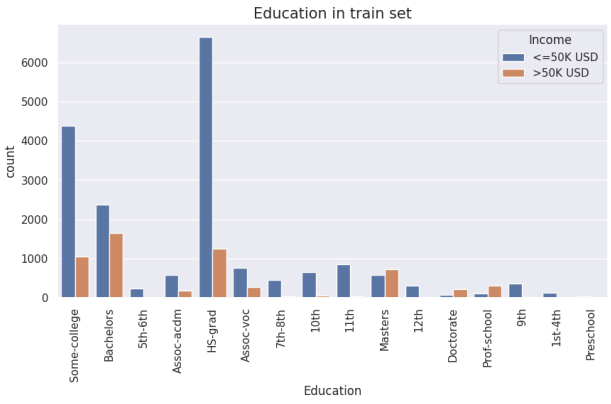

Fig. 14. Education in train set

- This figure suggests most higher income groups are married. Whereas lower-income groups are mainly un-

married, divorced or separated. This clearly conveys the message that income really influences marriage life.

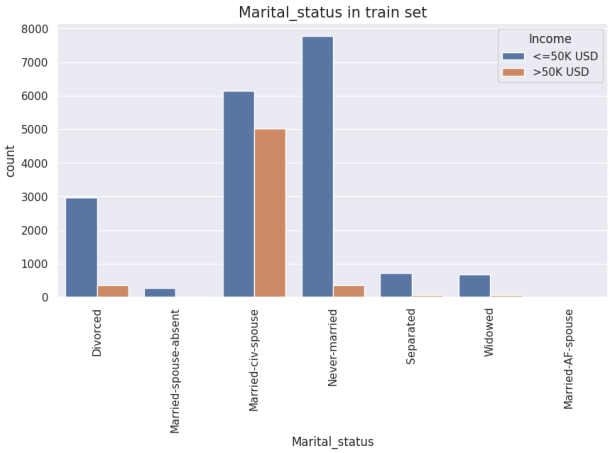

Fig. 15. Marital status and income count

- Already in the exploratory analysis, we mentioned that Executives and managers are the highest income groups. In the train set, we see the same too. Low-income groups mostly belong to clerics, other job workers and craft repairers.

Fig. 16. Occupation count in train set

- This plot gives us information regarding the relation between income and a person's race. In this case, it's somewhat difficult to draw an inference; although the highest income group is white, the largest number of lower income groups also belong to the white group. And the difference between these two is quite high.

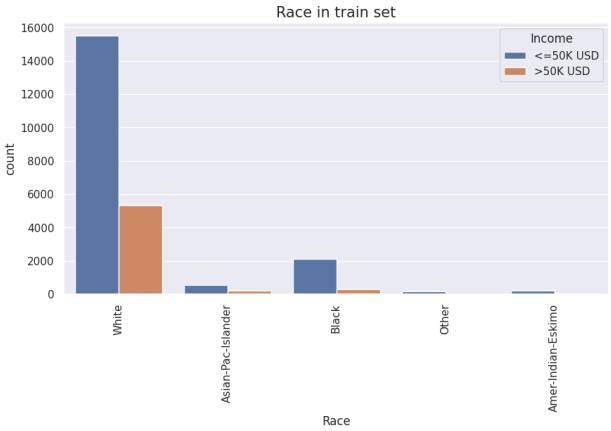

Fig. 17. Train set Race count

- Males are more likely to earn money than females. The higher income group is also from the male gender.

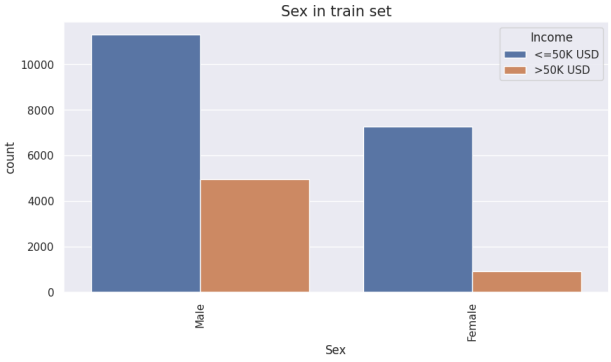

Fig. 18. Gender and income relation in train

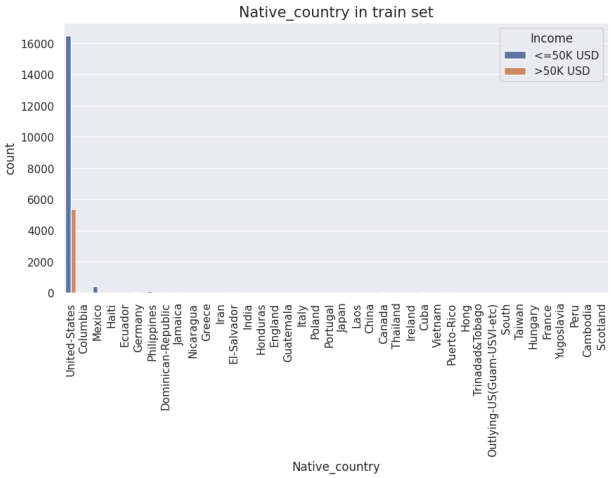

Fig. 19. Origin country and income relation

- People from the United States are more income groups. For other countries that didn't even appear in the plot, most have count values below 100. That's why they are

almost invisible in the plot compared to the count from the US.

2) Insights in testing data: We expect similar patterns in the predicted test data set. This will ensure the model is behaving consistently both on seen data as well as on unseen data.

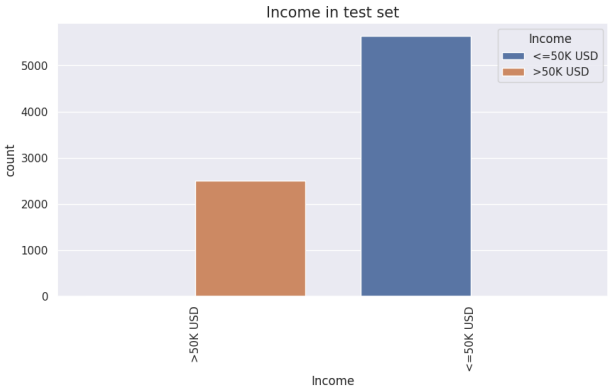

Fig. 20. Income count in test set.

In the prediction on the test data, we obtained 5634 data with below 50K income and 2507 data with above 50K income. The distribution looks almost the same as the original data in Fig 11. This is because we split the data in a stratified manner, which nearly retains the ratio of majority to minority class in train and test sets. This is a typical strategy followed in highly imbalanced classes. This technique ensures that the minority class is present in both the train and test sets during random shuffling and splitting of data.

- Just like in Fig 14 here also we see Bachelors, Masters and Doctorates are higher income groups. In fact the count for above 50K is more than below 50K. This is just because of the splitting nature of the data. Otherwise, the overall distribution is almost the same.

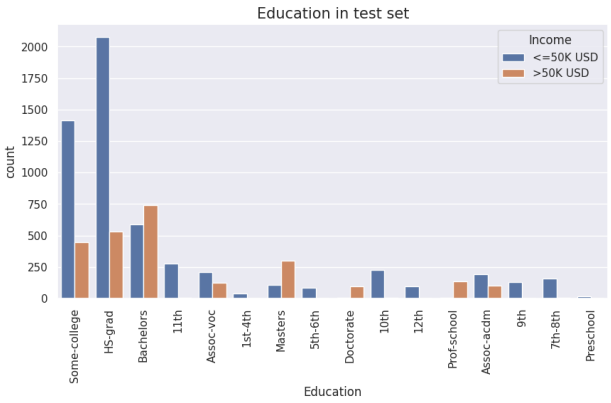

Fig. 21. Test set education distribution with income.

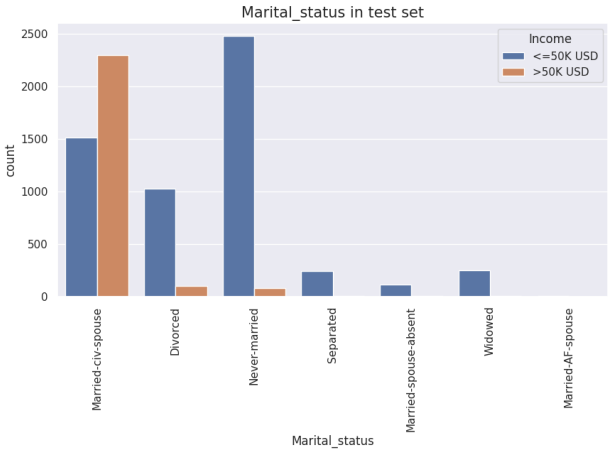

Fig. 22. Marital status on the test set

- Again, here, too, people who earn more have a higher frequency of marriage. The largest lower-income groups are those who never married; it may be possible that these people are school/ college students. Or maybe people who have lower earnings never opt for marriage. So overall, we see a similar structure in the model predicted set also. These are clear indications of the good performance of the model.

Fig. 23. Occupation in the test set

- Similar to the train data Fig 16, here, the high-income group is from executive and managerial posts. Further, here, a new occupation emerged as the highest income group Prof-speciality. If we look at the train data plot, this group had a fairly high number of people earning above 50K USD. In the current case, it may be more than below 50K class because of the test data-splitting nature. Again, this plot resembles the train plot quite well.

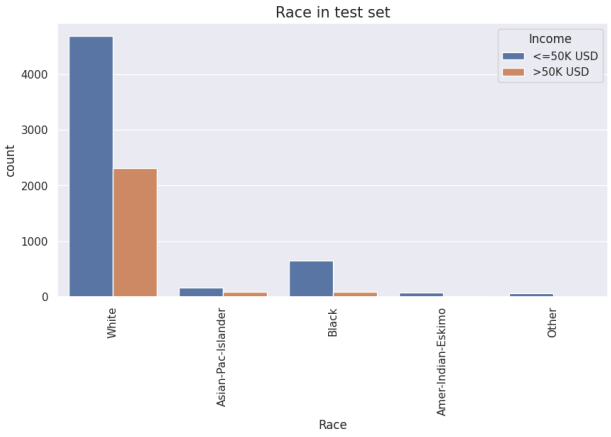

Fig. 24. Race in test set.

- Just like the one from train data here too, we are unable to draw a satisfactory conclusion. It is quite obvious that a person's race/ background should not affect his income potential. The income potential should be solely dependent on the education qualification, experience and exposure in a bias-free society.

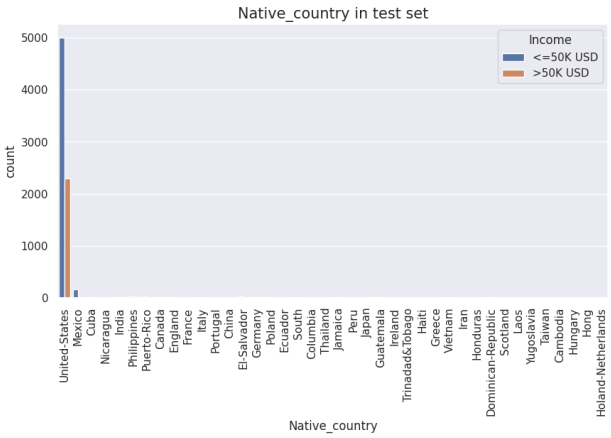

Fig. 25. Country vs income distribution in test set.

- This country distribution plot also looks the same as the training one, with people from the US being the largest in number for earning groups.

So, by observing the training and testing data, we can infer the following in terms of annual earnings:

- Education qualification plays a key role in deciding someone's earning potential.
- Executives and managers in private companies, professors and doctorates in education earn more.
- A person's earning really influences his/ her social life and relationships with others.

So, whatever inference we drew so far is from the train and model predicted test set; the behaviour of the predicted test set is quite similar to the train set ground truths. So, instead of relying only on the result of classification metrics, we cross-validated the entire model performance on our own, which gives good confidence in the working ability and explainability of the model.

V. Conclusions

Naive Bayes is the simplest and easiest algorithm to implement among various classification algorithms. It is quite easy to explain, unlike other models. In this assignment, we demonstrated the techniques of handling tabular categorical data and fitting a classification model using Naive Bayes. This assignment allowed us to explore the mathematical details of Naive Bayes and other statistical techniques. The name is consistent with the working process of this algorithm as it extensively uses the Bayes Theorem. In fact, we gained little insight into how the Bayesian framework works, unlike most conventional frequentist approaches for the data. There are certain limitations in the Naive Bayes, such as it assumes linear independence between the input variables, which is wrong in most real-world cases. In such a scenario, if someone still wants to use this classifier, they have to conduct a rigorous study of the data as we did in this assignment in the pre and post-training phase. If all results are satisfactory and meet the expectations, then only we can release such a model for production. Naturally, domain knowledge is the key for someone to do this kind of study instead of relying completely on the numbers and figures thrown out by the classification metrics. A Naive Bayes classifier is much simpler and less training expensive than logistic regression and neural networks. In many cases, Naive Bayes has proved superior performance than any other classification model. From this assignment, we understood that factors like education qualifications, nature of the job, and job sectors are the primary attributes that decide the earnings of a person. Earning, in turn, decides the social relationship of a person with others. Overall, this assignment is a great foundation for someone to begin working on Naive Bayes classifier-based applications.

References

[1] Naive Bayes: https://www.geeksforgeeks.org/naive-bayes-classifiers/
[2] Evaluation Metrics: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/
[3] Pandas: https://pandas.pydata.org/
[4] K modes: https://www.geeksforgeeks.org/k-mode-clustering-in-python/
[5] K prototypes: https://kprototypes.readthedocs.io/en/latest/api.html
[6] Scikit learn complement naive bayes: http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html
[7] Seaborn: https://seaborn.pydata.org/index.html

Access the original code here