AI for Social Good: Naive Bayes on the 1994 United States Census Data to Identify Income Inequality

Author(s): Jerry Tang

Stanford University CS109, Winter 2022

Abstract: Social inequality is magnified with recent rapid economic developments. As a result, it is more difficult for social scientists to keep up with the understanding of the impacts of such developments on humans as a whole as well as for specific demographics. The recent development of artificial intelligence (AI) and machine learning (ML) provides tools that are experts at finding the causation relationship between variables. Here, we propose a AI for social good application based on the Naive Bayes ML algorithm. The aspiration for the model is that it can be extended to most accurately predict important lifetime attributes (such as wealth, health, etc.) of an individual, given attributes of their background (such as race, gender, family background, etc.), as well as finding patterns and causes of social inequality through probability and machine learning. Here, we constructed a demonstration model with three input variables (race, sex, and native country) and mid-career income as the output. We trained this model using 10,000 data points from the 1994 United States Census and tested the model accuracy using 1,593 testing data points from the same source. We find that white males native to the United States is heavily favored to earn a higher annual income of $>$ \$50k, and that the income hierarchy very well matches today's expectations. The demonstration is a showcase of ML applications for social good with a high ceiling to explore.

Cover Image: from NeurIPS 2019

1 Introduction

Artificial intelligence is one of the fastest developing fields in both research and application[4]. With our society developing socioeconomically at a rapid pace, the application of artificial intelligence (AI) for the benefit of the society as become an emerging trend[9, 1]. This paper attempts to answer the question: how does your background at birth define the trajectory of your life? Consider two individuals, one born wealthy and one born poor. Which person is more likely to become wealthy by the end of their lifetime? One would be suspected to answer "yes". However, there are less-straightforward variables (short vs. tall, English speaking vs. non-English speaking etc.) that might have an impact on a person's life trajectory but are hard to predict but effectively presented through a machine learning algorithm over data sets. The ultimate proposal of this study is that we can add enough variables and training data to realistically predict the likely outcome of a person's lifetime (and the inequality that might be a burden for the people of similar demographics). Furthermore, we can take out insignificant data to reduce the weight of the model, or add more sensitive variable that can improve the accuracy of the model. Lastly, the significance of this model is to give social scientists the tool to understand the significant contributions to inequality, and from here, propose solutions to improve the well-being of the society in general.

We will provide a demonstration model by running a Naive Bayes[6][5] machine learning algorithm. We obtained data from the 1994 United States Census that includes various discrete and continuous variables. We will choose "race", "sex", and "native country" as inputs to predict whether the individual will achieve mid-career income of $>= \$50k$ per year. We will do so by training the Naive Bayes model on 10,000 training data points (that are each weighted differently to represent the density of such population the data point represents) and then perform accuracy test for our model using 1,593 testing data points from the same source that is excluded from the training data. Lastly, we will dump one sample of each demographic from all possible groups to see what the model's bias is for predicting their mid-career income.

2 Methodology

1. **Application Theory of Naive Bayes Machine Learning**

   This artificial intelligence application aims to predict specific likely outcomes of a person's lifetime if given personal background information. Mathematically, the model aims to do the following: given inputs $\{x_1, x_2, ...x_n\}$, for which each x_i is a random variable that represents a piece of background information about the individual, and output a value(s) from random variable(s) $\{y_1, y_2, ...y_n\}$ that represents event outcomes. Some hypothetical input information x_i we can include are: race, gender, family wealth, location of birth, or even information such as surrounding weather, height, or first language etc. Some hypothetical output information can be level of education, wealth, health, height etc. Such model would be pre-defined by the type of random variable each x_i and y_i is, and the machine learning process uses training data to obtain best estimations for the parameters of the random variables[2]. Once parameter estimations are obtained, the model can then take sample inputs and produce outputs accordingly.

   To obtain parameter estimations we will either use maximum likelihood estimation (MLE) or Maximum A Posteriori (MAP). The formulas are as follow, including the logarithm functions for ease of computer calculations:

$$\hat{\theta}_{MLE} = argmax_\theta f(x^{(1)}, x^{(2)}...x^{(n)}|\theta) = argmax_\theta(\sum_{i=1}^{n} log(f(x^{(i)}|\theta)))$$
$$\hat{\theta}_{MAP} = argmax_\theta f(\theta|(x^{(1)}, x^{(2)}...x^{(n)}) = argmax_\theta(log(g(\theta)) + \sum_{i=1}^{n} log(f(x^{(i)}|\theta)))$$

Here, $\hat{\theta}$ represents the best guess we can have for output random variables y_i that would make the input events the most likely (for MLE) or is the most likely outcome event given events $\{x^{(1)}, x^{(2)}...x^{(n)}\}$ (for MAP). The application difference between using MLE vs. MAP is that MLE is more prone to over-fitting while MAP is dependent on some imaginary prior belief.

However, since $\{x_1, x_2, ...x_n\}$ must not be perfectly independent in nature, a joint probability table is required to complete the exact calculations in both parameter estimation methods if we use the Brute Force Bayes ML method - if we define m as the size of set $\{x_1, x_2, ...x_n\}$, the size of such joint probability table is at least O(2^m) (simplest case of all Bernoulli's for x_i). To avoid this size problem, we introduce the Naive Bayes Assumption:

$$P(x^{(1)}, x^{(2)}...x^{(n)}|\theta) = \Pi_i^n P(x^{(i)}|\theta)$$

We assume the input conditions are independent and use the "and" rule of probability. This would greatly reduce the time and space complexity of the algorithm. However, it is worth noting that this is a blunt assumption and can be negatively consequential. For example, let x_1 be random variable that indicates a person's level of education and x_2 be the ethnicity of the person, yet these two variables are clearly correlated according to research[8]. This is not to claim that it would make the model invalid, but we would strongly suggest checking for strongly correlated sensitive data that might make the model heavily skewed.

2. **Demonstration Model: Methodology and Datasets**

We will build a Naive Bayes model to predict whether the individual has an mid-career income of more than $50k, given the race, sex, and native country of this person. Both the training and testing dataset come from the 1994 United States census, retrieved from `https://archive.ics.uci.edu/ml/datasets/Adult`. We choose to limit the scope to mid-career income because it would not make much sense to compare income between different stages of people's lives (for example, younger people usually earn less, and some older people might have retired)[7]. Furthermore, mid-career income is somewhat of a consistent qualifier for the well-being of a person's life, as this age bracket has the highest earnings[7]. The income distribution by age in the United States is illustrated in Figure 1. Here, we define this age group as those from age 40 to 59, and we will omit data points from other age brackets. For the Naive Bayes model, all four input and output variables are present in the dataset, but race and native country are not defined as binary variables. For the sake of simplifying the model, we make race into a binary varying that supports "white" and "not white" and native country a binary variable that supports "United States" and "other countries". Sex is given directly as either male or female, and income is given as $<= 50k$ or $> 50k$. Thus, the modified dataset has list $\{x_1, x_2, x_3, y\}$ for each row, representing the race, sex, native country, and income for some individual whose age is [40, 59] at the time of the census. We pick 1 to represent white, male, United States, and income > 50k, and 0 to represent other race, female, native country is not the United States, and income <= 50k.

There are 11,593 individuals (represented by a row) who satisfies the age bracket criteria. However, each row does not represent a single individual (or else you would need 300 million rows for the population of the United States). Instead, each row has a weight number, representing the number of people who is categorized as fitting into this row. Therefore, when counting for estimates in the model, the weight of each row must be taken into account. For example, if a row has weight 10, this means we can treat it as 10 identical rows with same output for each variable. Similarly, we will account for weights in the testing data when computing for model accuracy.

We divide the 11,593 rows of representing individuals into two sets: a training set and a testing set. The training set will include the first 10,000 rows, while the testing dataset will include the last 1593 rows. The choice is not purposely optimized, but later results will show that this partition makes little difference to the model accuracy.

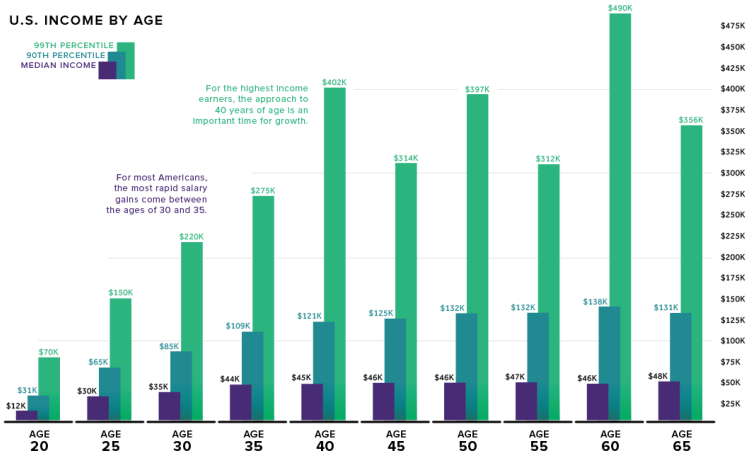

Figure 1. Income based on age groups. We see that people earn the most in their life time between age 40 to 60. We define this age bracket as "mid-career" and will run Naive Bayes classification for demographics within this age bracket.

3. Demonstration Model: Implementation

We implement a Naive Bayes training and testing model using Python 3. The outline is as follows:

(a) Import dataset and filter to leave only data that matches the age bracket we want.

(b) Separate the first 10,000 rows and last 1593 rows of the dataset into the training data and testing data. Only take the four columns of random variables we want, and hash them into 1's and 0's based on our binary classifications. Write into seperate CSV files.

(c) Make an array that records the weight of the first 10,000 rows, each corresponding the the same index in the training dataset.

(d) Condition on y = 1 or y = 0, and count how many occurrences of successes (1) and failures (0) for x_1, x_2 and x_3 as well as how many counts of y = 1 and y = 0.

(e) We choose to use MAP with Laplace assumption as our method of parameter estimation. Generate complete MAP estimations between y = 0, y = 1, and all x_i using the following formula (note that we are using the Bayes Assumption here, so there is no need to obtain the complete probability table:

$$p(x_i = a|y = b) = (\#\text{ of training examples s.t. } x_i = a \text{ and } y = b \text{ plus } 1) \ / \ (\#\text{ of training examples s.t. } y = b \text{ plus } 2)$$

From which, the following MAP estimates are obtained:

y	$x_1 = 0$	$x_1 = 1$
0	0.179	0.821
1	0.104	0.896

Table 1. Laplace MAP estimates for y and x_1

y	$x_2 = 0$	$x_2 = 1$
0	0.383	0.617
1	0.129	0.871

Table 2. Laplace MAP estimates for y and x_2

y	$x_3 = 0$	$x_3 = 1$
0	0.127	0.873
1	0.080	0.920

Table 3. Laplace MAP estimates for y and x_3

(f) The model is finished training. We can feed in tuples $(x_1 = j, x_2 = k, x_3 = w)$ for $j, k, w \in \{0, 1\}$ as testing points

Last updated 3.11.2022: The exact data-set and python code can be found at my github page or `https://github.com/sourpassionfruit/Challenge2.0`. The github folder should contain everything needed to just click run and replicate the results of this study.

4. **Model Verification** Testing data is inputted into the model and compared with the known result. The accuracy of the model is given by the following formula:

$$\text{Accuracy} = (\text{number of correctly classified testing data}) \,/\, (\text{number of total testing data})$$

When counting the correctly classified inputs and the total testing, we also need to account for the weight of each row/data point. This way, individuals are being represented close to the real world proportions.

3 Results and Further Testings

The intermediate results (MAP table) for our demonstration model has already been given in the previous section. The accuracy of the trained model is the output:

```
100%|██████████| 10000/10000 [00:00<00:00, 482903.20it/s]
100%|██████████| 1593/1593 [00:00<00:00, 736499.81it/s]
Number of training data (rows, before multiplying by weights) is: 10000
Number of testing data (rows, before multiplying by weights) is: 1593
Accuracy is: 0.6135943674902062
```

Figure 2. Testing samples with Naive Bayes model trained with 10,000 training examples. For list (x_1, x_2, x_3), x_1 represents the wealth the race of the person, and x_2 represents the sex of the person, and x_3 represents whether the person's native country is the United States. Model accuracy reflects inequality in income between demographics. The accuracy should be 50% for complete distributive fairness (can't predict based on race, etc.).

We can do a very naive optimization for the result of the accuracy by finding the balance between the number of training data and the number of testing data. This is because there is a trade-off between having sufficient training data to train the model and having sufficient testing data to be consistent with the accuracy result. Interestingly, it was observed that 10,000 training data rows is far more than sufficient training data, and so is 1,593 testing data rows. Significantly scaling towards either end does not alter the accuracy by a significant margin. This result is presented in table 4. However, in theory, a larger set of training data is preferred.

Training data size	Accuracy
100	0.612
1000	0.618
5000	0.615
10000	0.614
11000	0.604
11500	0.645

Table 4. Training data size does not impact the accuracy. Any choice from the table above is sufficient training data and testing data to yield consistent predictions.

From an ethical standpoint, if complete distributive fairness is the case for income between the demographic groups we selected, the accuracy of the model should be ∼0.5, indicating that we cannot know the income of a person by judging on the person's race, sex or country of origin. Instead, we have ∼0.6 accuracy. From one perspective, 0.6 is not very deterministic, but on the other hand, a clear deviate from 0.5 indicates that there is indeed a consistent trend of inequality. However, it is worth noting that a inherent weakness of Naive Bayes is that it ignores dependency between variables. In this case, country of origin and race might very well be correlated and proved or disproved (tho this study will not statistically prove it).

We also tried to slightly change the definition of variables to see if the model is more biased towards a specific group. We changed the race variable to "Black" and "Others", replacing "White" and "Others". This also yields very similarly accurate model results (0.618 vs. 0.614), which is to say there is likely no significance in altering this variable. We did the same isolation experiment for all other racial groups including "Asian-Pac-Islander", "Amer-Indian-Eskimo" and also found no significant change in accuracy (0.615 and 0.618, respectively). We also did the same for native country and also cannot alter the accuracy of the model by any significant amount (>0.01).

Once the model is trained and tested, we will throw in every combination of (x_1, x_2, x_3) into the model trained based on the original variable settings. We will also generate a "confidence score" for each of the outcomes. Recall that the Naive Bayes outcome is dependent on whether the cumulative parameter estimation (across all x's) is greater for y = 1 or y = 0, we call them L1 and L0. If L1 = L0, the model is not as confident in determining y = 0 or y = 1 as say, L1 - L0 = 100, which indicates that Y = 1 is much more optimal, or L1 - L0 = -100, which indicates that Y = 0 is much more optimal. The algorithm goes as the follows:

1. Generate possible inputs, total size is 2^3
2. Run each input into the trained model along with the addition of a confidence score. Print the results for each outcome.

, and the result is:

```
This person is White Male from United States, and income is > 50k. Confidence score for this predicion is 0.004304895357839511
This person is not white female from not the U.S., and income is <= 50k. Confidence score for this predicion is -2.5766841369833724
This person is not white female from United States, and income is <= 50k. Confidence score for this predicion is -2.0639472065298934
This person is not white Male from not the U.S., and income is <= 50k. Confidence score for this predicion is -1.1356921367460648
This person is White Male from not the U.S., and income is <= 50k. Confidence score for this predicion is -0.5084320350956397
This person is White female from United States, and income is <= 50k. Confidence score for this predicion is -1.4366871048794676
This person is White female from not the U.S., and income is <= 50k. Confidence score for this predicion is -1.9494240353329473
This person is not white Male from United States, and income is <= 50k. Confidence score for this predicion is -0.6229552062925854
```

Figure 3. The trained Naive Bayes model only predicts a person's mid-career income to be >$50k if the person is White, Male, and born in the United States.

The result indicates that, based on the Naive Bayes model trained on the 1994 United States census data, a person will only be predicted to have a mid-career income of $> \$50k$ if the person satisfies 1) White 2) Male 3) born in the

United States. This is a unsurprising result especially considering the time period, but shocking nonetheless on how accurately the Naive Bayes model identified the most privileged population demographic.

Recall that a very high confidence score indicates that the model strongly believes in the person having an income of > $50k, and a really small negative score indicates that the model strongly believes in the person having an income of <= $50k. The confidence scores matches the decision of the model - it is only in the scenario where the person is White, Male, and from the United States that this score is positive (or else won't return such in the Naive Bayes classification process). However, it is a very small positive number - indicating that this result might be less deterministic. Actually, simply training the data on slightly different dataset (for example the first 5,000 rows of data rather than 10,000) would yield -0.002, which would also categorize this demographic as more likely to have income <= $50k. Nonetheless, even in that case, the absolute value of the negative confidence scores can tell us about the inequality. If we were to rank the demographic groups from highest confidence score to lowest, it would be:

White/Male/U.S > White/Male/not-U.S. > not-White/Male/U.S > not White/Male/not-U.S. > White/Female/U.S > White/Female/not-U.S. > not-White/Female/U.S > not-White/Female/not U.S

This sequence can indeed be seen as a rough estimate for the rank of privilege in terms of income between demographics. For example, non-white females from outside of the U.S has strongest belief by the model that she will be poor. Again, the results are unsurprising but shockingly logical and matching to other proven sources on income inequality[3].

4 Conclusion: For Social Good, a Glorious Aspiration

Let's first summarize the study's key methodology and findings: we trained a Naive Bayes model using MAP-Laplace and based on 10,000 data points and produced an accuracy of 0.614 for the 1,593 testing data points, all from the 1994 United States census. We then identified that white males who are born in the U.S holds much higher privilege when it comes to income inequality, and that the ranking of most-privileged to least-privileged is very well matches our impressions today.

It is easy to think of AI as anti-human and hard to think of applications of AI to more than technical problems and also society issues and challenges to the development of humanities. Here, we demonstrated how a simple machine learning algorithm like Naive Bayes can have surprisingly strong applications. Nonetheless, the model and dataset are not perfect and ethnically is negligent of many facts: 1) Naive Bayes assumes that variable are independent when they clearly are not (for example race and country of origin), which will skew the data 2) for simplicity, we chose to define model variables as Bernoulli variables, but this is ignoring the minority data such a) non-traditional gender b) difference in ethnicity, not just race c) the choice of neglecting income other than "mid-career" age is an assumption on the overall trajectory of a person's career path but is a over-generalization. Nonetheless, the goal of AI application in social goods should be to continue pushing the boundary of what scenarios are AI ethnically and technologically suitable.

Lastly, this project was originally planned to be a much more extensive research project (Yes, just for CS109 Challenge, and even better than this). Here are some original intents I regret to have not achieved due to time and knowledge limitations.

1. Including all random variables from the 1994 census dataset, including non-Bernoulli and continuous random variables.

2. Find correlation between input variables and "cut" the model by removing insensitive data

3. Add other sources of sensitive variables

4. Find a dataset with non-binary classification output and with more than one output variable

5. Most importantly (and what I really wanted to do from the beginning, as well as in the future), use of neural network and deep learning to connect the likely events of a person's lifetime. For example, if a person is born with {X} conditions and is predicted to {go to college, not have kids until age 30, join the military}, how would that impact the next layer of events in the person's lifetime? And so on...

When the opportunity presents, the project will be expanded based on these ideas. I want to give special appreciation to Chris Piech and his teaching team. Examples in CS109 scream "for social good" and absolutely inspires students to think of computer science in application with societal problems. AI for social good - what a glorious aspiration.

References

[1] Neurips joint workshop on ai for social good workshop at neurips2019.

[2] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. *Machine learning*, pages 3–23, 1983.

[3] Fernando G De Maio. Income inequality measures. *Journal of Epidemiology & Community Health*, 61(10):849–852, 2007.

[4] Sreetama Dutt, Anand Sivaraman, Florian Savoy, and Ramachandran Rajalakshmi. Insights into the growing popularity of artificial intelligence in ophthalmology. *Indian Journal of Ophthalmology*, 68(7):1339, 2020.

[5] Kevin P Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18(60):1–8, 2006.

[6] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[7] Nick Routley. Visualizing american income levels by age group, Mar 2019.

[8] Camille L Ryan and Kurt Bauman. Educational attainment in the united states: 2015. 2016.

[9] Nenad Tomašev, Julien Cornebise, Frank Hutter, Shakir Mohamed, Angela Picciariello, Bec Connelly, Danielle Belgrave, Daphne Ezer, Fanny Cachat van der Haert, Frank Mugisha, et al. Ai for social good: unlocking the opportunity for positive impact. *Nature Communications*, 11(1):1–6, 2020.