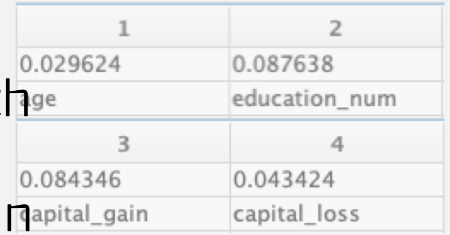
1. Intro

Socio-economic Signals to Income : Which Are Important Signals and How to Predict Income

1. We used UCI 'Adult Data set' (http://archive.ics.uci.edu/ml/datasets/Adult) which shows 14 socio-economic indicators such as marital status, profession and corresponding income in binary form('>50k','<=50k'), having about 50,000 instances.

2. (1).We tried to do Linear Regression from 14 socio-economic indicators to the binary values of '>50k', '<=50k', to know which factors are given which weights. By looking into the weights the Linear Regression model produces, we could know which signal is considered important in predicting income. (2). Then, we tried logistic regression to make a prediction model to tell us whether a person of certain values in 14 indicators is highly likely to earn that income or not. If the data is well-clustered in a binary sense, this simple model will function properly. (3). Finally, in order to make more subtle classification model, we tried Deep Neural Network classification to boost the accuracy in classification.



1	2
0.029624	0.087638
age	education_num

3	4
0.084346	0.043424
capital_gain	capital_loss

3. We thought this is interesting because :
(1). If we apply this examination into many other years' census data, we could track the change of the importance given to each indicators signaling income in timeline.
(2). The similar linear regression and prediction model approach could be applied practically in welfare policy making and target marketing.

2. Dataset

1. Original Dataset

The original data comes from 1994 Census database of America. It contains 14 attributes : age, workclass, fnlwgt*, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital loss, hours-per-week and native-country, given in string category data or integers. Then it shows whether the person makes over $50K a year or not. The data contains total of 48842 instances.

*fnlwgt shows the number of people the census believes the entry represents. We didn't use this column because we thought this was irrelevant in our research.

2. Data Preprocessing

First, we divided non-sequential and string-categorical columns to separate columns that each represent the whole categories that belong to that original column. We also singled out 'null' column to prevent data loss(14 columns and omitting the null-including instance was fatal). For instance, occupation column was divided into 14 independent one-hot columns.

Second, integer valued columns(age, education level, weekly working hours) were z-score normalized for effective gradients update for every column. However, there were the integer valued columns that did not have normal distribution but rather have the distribution similar to that of power law. Capital-gain and capital-loss column were such columns. For those columns we first scaled them down by log of base 10 and then applied z-score, to counter the different distribution. Fortunately, these columns didn't have undefined values.

To sum up, as a results of dividing string category columns, our dataset matrix came to have 91 columns with the same 48842 instances.

3. Algorithms

1. Linear Regression

Our Linear Regression Algorithm accepts 91-dimension array(including one dimension for bias) as an input and tried to find a set of weights that minimizes the 'the average of squared residual' between batch datapoints and the labels. We used Gradient Descent to optimize the weights. (Compared this with the weight solution of Matlab ' \ ' to get the sense of goal accuracy of linear regression).

2. Logistic Regression

Our Logistic Regression model accepts 91-d array as an input and try to find a set of weights that makes up boundary that minimizes the following error function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$

We used 'Gradient Descent' to optimize the set of weights.

3. Neural Network

The network includes 1). Input Layer – 2). Fully Connected Layers – 3). Softmax Layer. The built-in 'trainNetwork' function was used to train the network. After the network was trained, the validation data was classified.

4. Results (1). linear regression

Our Linear regression showed the average of 83% accuracy for testing dataset. It peaked in the epoch 7 and stopped growing in accuracy.

We compared our Linear Regression Model's performance to Matlab '\' solver to have the goal of the model's accuracy. The Matlab solver showed 84% accuracy in average, which is equivalent to our model's performance. We could conclude that our gradient descent successfully converged near to global minima.

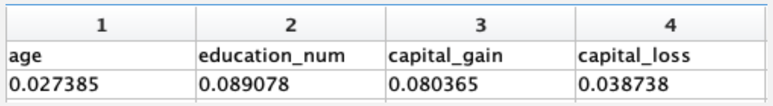
[Insights for original research purpose]

We wanted to know which factors are to be considered important in predicting income through research. We looked into trained weights that corresponds to each 91 dimensions of data.

We first singled out weights for z-scored integer columns which have range of approximately -3 to 3. The 'education level' and 'capital gain' were given about four times the weights of age or capital loss as the signals for having over 50k income in 1994. For government officials, the person working in federal government was more important a signal to predict earning over 50k$ than local or state government(The other col-weights information is included in [col-weight.mat]file).

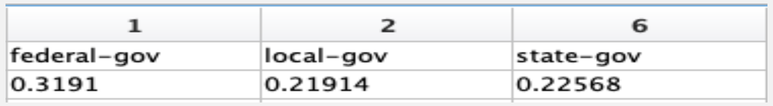
ex).

z-scored indicators



1	2	3	4
age	education_num	capital_gain	capital_loss
0.027385	0.089078	0.080365	0.038738

non-z-scored indicators



1	2	6
federal-gov	local-gov	state-gov
0.3191	0.21914	0.22568

4. Results (2). logistic regression and deep neural net

Our Logistic Regression also showed the average of 84% accuracy in classifying over 50k and under. It peaked the accuracy since 5th epochs and did not show improvements afterwards.

Our neural network showed the average of 77% accuracy for validation data. We tried out different structure of layers and many epochs, but the accuracy did not seem to improve much.

Han Seokhee (2013130874)
Chung Hyelee (2017130776)
Hwang Jongho (2018320177)