# Income Prediction Using Supervised and Unsupervised Learning

Samir Khanal
Department of Computer Science
Lambton College
Toronto, Canada
samirkhanal35@gmail.com

0000-0002-1778-2374

***Abstract*—As the title of the paper, the research is conducted to predict whether the persons' income exceeds $50K/year or not. The input data about the different factors are collected from a given UCI dataset which is based on census data. The dataset is then processed through data cleaning, handling of missing values, Outlier removal, feature selection, data normalization, data visualization, and splitting of data into a training set and testing set. The training set is then fitted into Supervise Learning model, Gaussian Naïve Bayes model, and into the Unsupervised learning model, KMeans from scikit learn library. The Gaussian Naïve Bayes model is tested with the testing dataset, whereas the optimal number of clusters is calculated in KMeans**

**Keywords—Income, Missing Values, Outliers, Feature Selection, Normalization, Gaussian Naïve Bayes, KMeans**

## I. INTRODUCTION

People of different ages with different educational backgrounds and different years of experience have different amounts of income. The amount of income is dependent on different factors. We can predict whether the persons' income is greater than 50K or not based on different factors.

## II. METHODOLOGY

### A. *Data Acquisition*

There is variety in peoples' income on different basis. Barry Becker extracted the data from the 1994 Census database and created an dataset named Adult data set which is currently found in UCI Machine Learning Repository[1]. This dataset was used for the prediction of income.

### B. *Data Preprocessing*

While checking the data, the target feature 'Salary' only contained two unique values: <=50K and >50K. So, its values were converted to 0 and 1 for salary <=50K and >50K, respectively.
Then the data was split into 70% of the training set and the rest into the testing set using the python array splitting method. We moved on to the next steps with train data.

1. Missing Value Handling

   The null or missing values are represented as '?' by the dataset collector. Three features, 'workclass', 'occupation', and 'native-country'. As all three features were categorical, all the data with missing values were removed. The duplicate datas were also removed.

2. Categorical to numerical

   The categorical columns were differentiated from the dataset, and their categorical data were converted into numerical data using a label encoder found in preprocessing module from scikit learn library.

3. Outlier Removal

   The data was visualized using histogram and box plot to determine its dispersion and the outliers.

   i) Using Median-IQR

   Features 'age', 'education-num', 'fnlwgt', and 'capital-gain' were analyzed for outliers by using Median-IQR method and the Outliers were removed.
   Here, the outliers are determined as:
   Q1: First Quartile, Q3: Third Quartile
   IQR = Q3 – Q1
   OutlierConstant = 1.5
   OLB = Q1 - (IQR * outlierConstant)
   OUB = Q3 + (IQR * outlierConstant)
   OUB < Outlier & Outlier < OLB
   Where, OLB: Outlier Lower Boundary, OUB: Outlier Upper Boundary

   ii) Using Mean-SDV

   Features 'Capital-loss' and 'hours-per-week' were analyzed for outliers by using Mean-SDV method and the Outliers were removed.
   Here, the outliers are determined as:
   OLB = mean – 3*std
   ULB = mean + 3*std
   OUB < Outlier & Outlier < OLB
   Where, std: standard deviation, OLB: Outlier Lower Boundary, OUB: Outlier Upper Boundary

4. Feature Selection

   The correlation between the features was calculated and rounded to the decimal point of two decimals. Heatmaps were created using the correlation data of all 15 columns. Four feature columns were selected by keeping the threshold of 20% correlation percentage. They are: 'age', 'education-num', 'sex', and 'hours-per-week'.

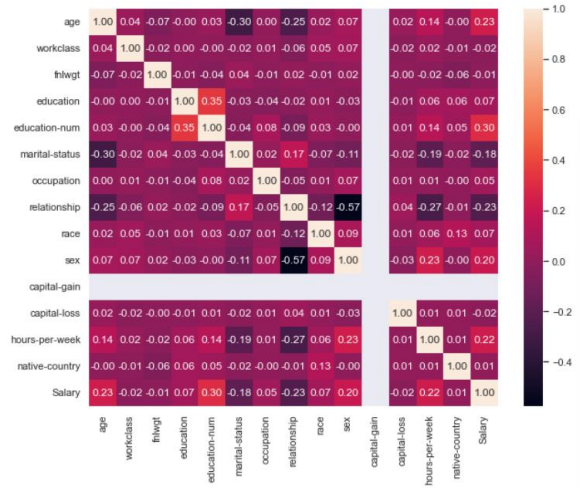



Fig 1: Heatmap of correlations between the features

## 5. Data Visualization

The distribution of the target variable 'Salary' was visualized. Then, the distribution of each selected feature was visualized with a histogram plot along with their central tendency by using a Normal distribution graph.


Fig 2: Distribution of target variable

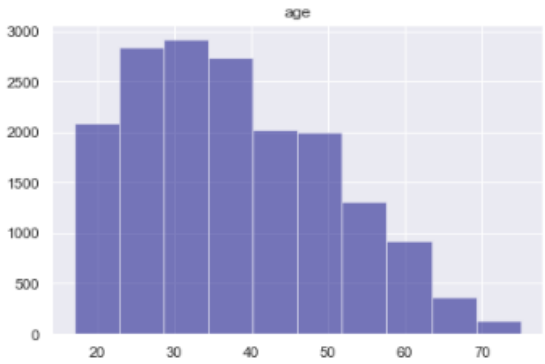



Fig 3: Data distribution of age

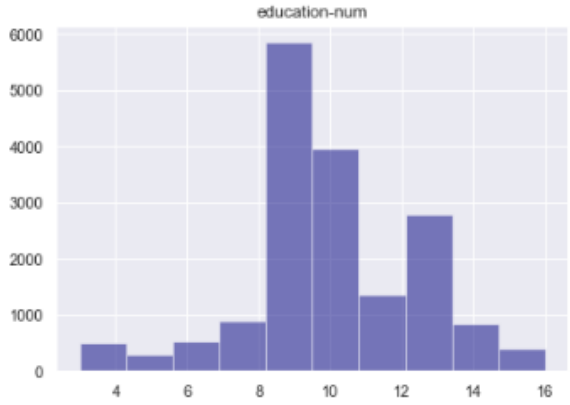



Fig 4: Data distribution of education-num

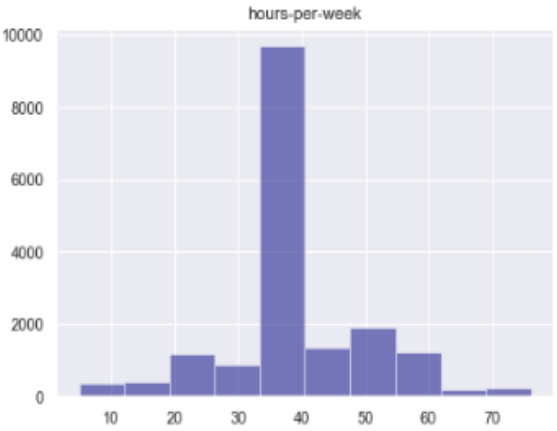



Fig 5: Data distribution of hours-per-week


Fig 6: Normal distribution of age

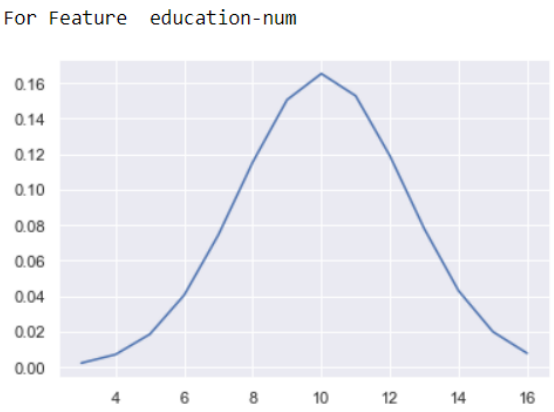



Fig 7: Normal distribution of education-num

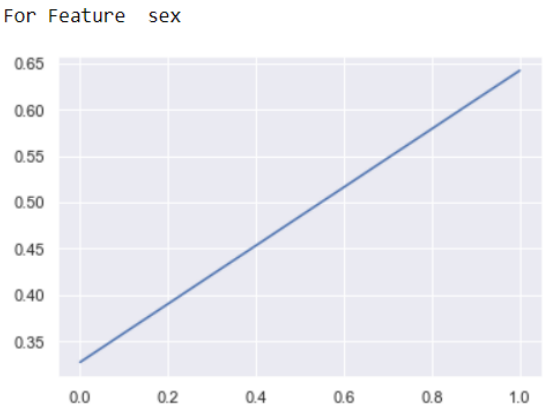



Fig 8: Normal distribution of sex

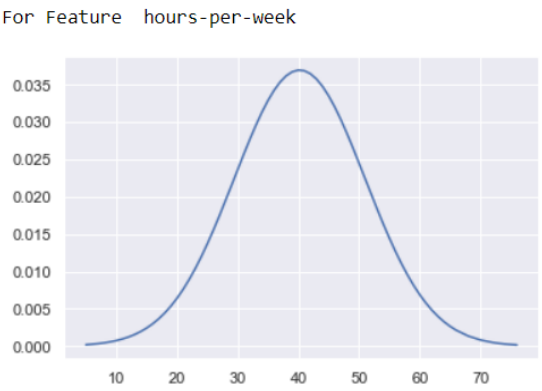



Fig 9: Normal distribution of hours-per-week

6. Data Normalization

To normalize the data; the scaling data normalization method was applied to the feature columns of the dataset using the formula in equation (1).

$$\text{Normalized value(x)} = \frac{x - \textit{minimum value}}{\textit{maximum value} - \textit{minimum value}} \quad (1)$$

## C. *Data Modelling*

1. Supervised Learning

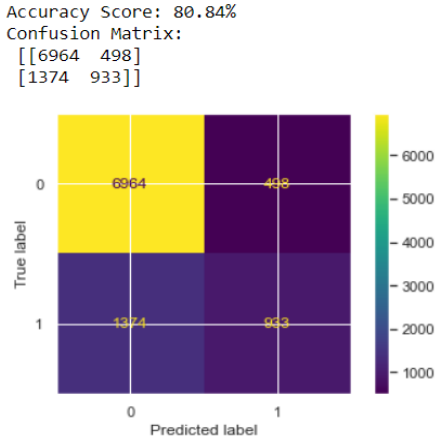
The target variable or dependent variable only had two categories of data 0 and 1. So, the binary classification technique, Gaussian Naïve Bayes model was used from Scikit learn library.

2. Unsupervised Learning

The target variable was removed from the data, and other selected features were fed to the KMeans model from Scikit learn library with different numbers of clusters.

## III. PERFORMANCE AND RESULTS

A Gaussian Naïve Bayes model from scikit-learn was used to fit the data. The model gave 82.65% accuracy to the training dataset and 80.84% accuracy for the test dataset. The confusion matrix of the model for the test dataset is shown below. It has 6964 True Positives and 933 True Negatives, 1374 False Negatives, and 498 False Positives.




| Classification Report: | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 0.93 | 0.88 | 7462 |
| 1 | 0.65 | 0.40 | 0.50 | 2307 |
| accuracy | | | 0.81 | 9769 |
| macro avg | 0.74 | 0.67 | 0.69 | 9769 |
| weighted avg | 0.79 | 0.81 | 0.79 | 9769 |

Fig 10: Model evaluation with confusion matrix

A KMeans model from scikit-learn was used to fit the data. We used different numbers of clusters to calculate the distortion and inertial of the model. In the elbow graph of both of them, the value of k at elbow is 2. So, the optimal number of clusters for this data is 2.

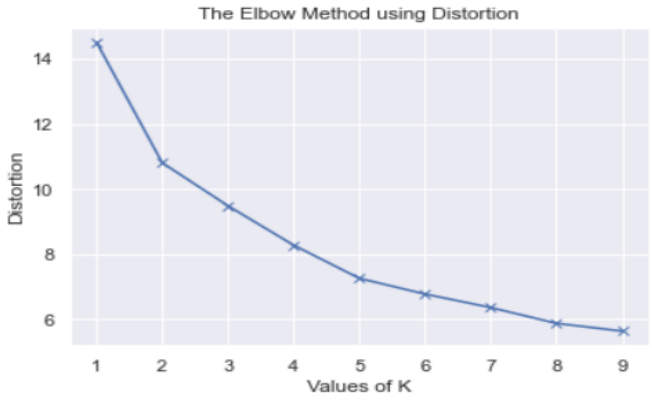



Fig 11: Elbow Method using Distortion Values

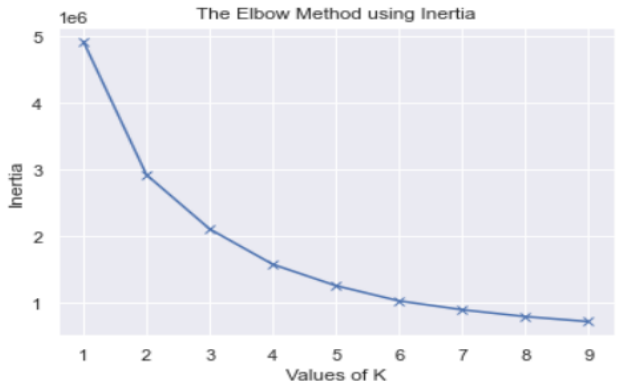


Fig 12: Elbow Method using Inertia Values

## IV. CONCLUSION

The two methods presented in this paper can process the data in Adult Data Dataset, clean data, deal with missing values and outliers, feature selection, normalization, visualization, modeling, training, and predicting the Salary whether less than 50K OR greater than 50K.

There is still some further research to do. For example, Other classification models can be used and compared with our models.

## ACKNOWLEDGMENT

This research article would not have been possible without the exceptional support of my mentor Prof. Mohammad Islam and the UCI dataset [1]. I am also grateful for the insightful comments offered by the anonymous peer reviewers from my batchmates. The generosity and insights of Prof. Mohammad Islam have improved this study in innumerable ways. Thank you all for your support.

## REFERENCES

[1] Kohavi, R. and Becker, B. (1994). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/adult

[2] Gupta, A. (22 August, 2022). Elbow Method for optimal value of k in KMeans. https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/

[3] Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. http://robotics.stanford.edu/~ronnyk/nbtree.pdf