Adult Income Based on Socio-Economic Status

**Data Mining**

**Project Report ITEC-4305 M - Web Mining**

2024 Winter

Submitted by:

Name: Mukta Dey

St id:216748469

Name: Jason

Student id:

Name: Teja

Student id:

Professor & Advisor: Soroush Sheik

**Index**

|  |  |
| --- | --- |
| Content | Page |
| 1. [Overview](#bookmark=id.gjdgxs) | 3 |
| 1. [Problem Statement](#bookmark=id.30j0zll) | 3 |
| 1. [Data Exploratory analysis](#bookmark=id.1fob9te) |  |
| * 1. [Variables](#bookmark=id.3znysh7) | 4 |
| * 1. [Missing Values](#bookmark=id.2et92p0) | 5 |
| * 1. [Unbalanced Data](#bookmark=id.tyjcwt) | 6 |
| * 1. [Comparison between income and other variables](#bookmark=kix.f0ftqkuiy9qi) | 6-10 |
| 1. [Feature Correlation](#bookmark=id.3dy6vkm) | 10 |
| 1. [Data Pre-processing](#bookmark=id.1t3h5sf) |  |
| * 1. [Data Encoding](#bookmark=id.4d34og8) | 11 |
| * 1. [Imputation](#bookmark=id.17dp8vu) | 11-13 |
| * 1. [Outliers](#bookmark=id.tl0doobpg40d) | 13 |
| * 1. [Normalization](#bookmark=id.3rdcrjn) | 14 |
| * 1. Feature Engineering | 14-15 |
| 1. [Algorithms](#bookmark=id.tfe7rllaum2y) | 15-17 |
| 1. [Evaluation](#bookmark=id.bzecw5qpls1q) | 17-20 |
| 1. [Conclusion](#bookmark=id.mtsrwx4rxlz1) | 20 |
| 1. [Python Code Link](#bookmark=id.ph0drq7fv8uq) | 21 |
| 1. [Team Collaboration](#bookmark=id.7jr9jpkfoyaw) | 21 |

1. ***Overview***

**Project Description**

Our data mining project uses the real-world dataset extracted from 1994 census bureau database to predict annual incomes for adults in the United States, given a set of attributes like employment details, demographic information etc.

The income level is classified in two classes – less than or equal to 50,000 (0) and greater than 50,000 (1). The code for this project is written in Python.

**Summary of results**

For our project, We choose 5 different models to check which model gives the best accuracy (from K-nearest neighbors, Random Forest, SVM, Logistic Regression and Decision tree). We checked which one provided the highest accuracy score. However, Random Forest came pretty close to the optimal model in terms of accuracy, actually yielding the best results for precision, F1-score, false positive rate, and true negative rate.

1. ***Problem Statement***

Using 1994 Census Bureau data, the aim is to build a predictive model that determines the income level for adults. Income level is a binary target variable that indicates whether an individual makes less than or equal to $50K or greater than $50K on an annual basis.

This project requires us to explore classification algorithms on a real-world dataset and write a report explaining our experimental results. The language of implementation can be anything — the only requirement is that our program be able to interpret the given data and be able to classify instances and produce interesting statistics.

The algorithm should be based on the classification algorithms learned during the course. Usually, a straightforward implementation of one method will not lead to satisfactory performance. Also, the algorithm can be a combination of methods and should incorporate one or more data mining techniques when the situation arises. These techniques include (and certainly not limited to):

* Handling imbalanced dataset
* Proper imputation methods for missing values
* Different treatment of various types of features: continuous, discrete, categorical, etc.

1. ***Exploratory Data Analysis***
   1. Variables:

Our census data contained 15 variables of three distinct types: continuous, categorical and ordinal. We only had one ordinal value, education, which was originally named education-num and described as continuous.

We deleted the original *education-num* variable, which was recorded in int format, opting instead to use its neighboring column, *education,* as the new *education* variable because it translated its neighbor’s int values into their corresponding String values. Since our new *education* variable consisted of a string whose order was determined by the level of education.

**Categorical Attributes**

* ***workclass***: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
* ***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.
* ***marital-status***: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
* ***occupation***: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
* ***relationship***: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
* ***race***: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
* ***sex***: Female, Male.
* ***native-country***: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad& Tobago, Peru, Hong, Holand-Netherlands.

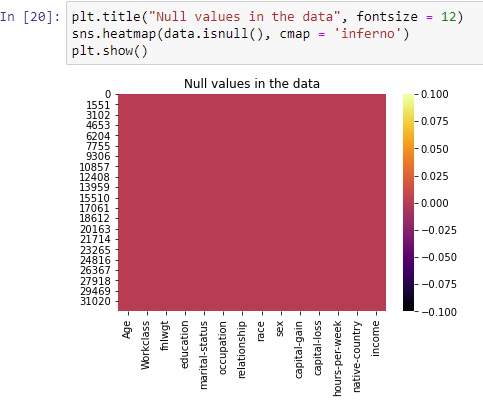
**Continuous Attributes**

|  |  |
| --- | --- |
| * **age** * **fnlwgt (final weight)** * **capital-gain** | * **capital-loss** * **hours-per-week** |

**Ordinal Attribute**

* **education-num (deleted)**
  1. Missing values:

We checked to see if there were any Null values present in the data that we have. Handling the null values was the first thing that we needed to do.

****

Then, we have a look at the data types, of the other features and the value counts and unique values in those features. Null values were as ‘?’ in the data. In our dataset out of the 48,842 entries, there are 3 columns have null values:

1. **Workclass** = 1836 null values
2. **Occupation** = 1843 null values
3. **Native-country** = 583 null values

Hence, we fix this with the most frequent element(mode) for the entire dataset. It generalizes well, as we will see with the accuracy of our classifiers in later sections.

3.3 Unbalanced data:

The data is imbalanced, it can cause overfitting and bias in the prediction. So, it is important to check and cure the data imbalance if present. We check the target variable to see if it is balanced or not.

Chart, bar chart

Description automatically generated

3.4 Comparison between income and other variables

Here we also look at the majority of people who have income more than 50K per year with respect to different attributes:

1. **Workclass**: We see that the majority of people who have income >50K a year are from the private sector. The same goes for people with income <=50K.

A graph of a bar chart

Description automatically generated with medium confidence

1. **Education:** We can see that the majority of people who have income >50K a year are people who have Bachelor's degree. While people who have income <=50K are from HighSchool-grad sector.

A graph of a number of students

Description automatically generated

1. **Marital-status**: Married people have a higher income as compared to others.

A graph of a number of people

Description automatically generated

1. **Occupation**: Majority of people whose income is greater than 50K are either executive managers or they belong to any professional specialty. A graph of a number of people

   Description automatically generated
2. **Relationship**: Husbands in the family have a higher salary as compared to other relationships in the family. A graph of a number of people

   Description automatically generated
3. **Race**: White people have a higher salary as compared to other races. A graph with blue and yellow squares

   Description automatically generated
4. **Sex**: We can see that male have more salary than female. Also in the dataset, the number of men is more than women. A graph of a bar chart

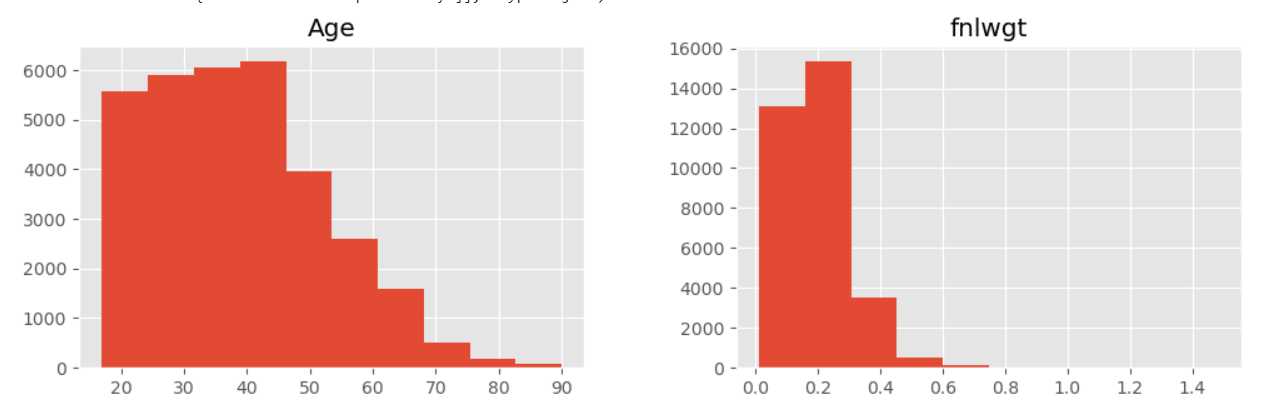
   Description automatically generated with medium confidence
5. **Native-country**: we can see that both the majority of people whose income is >50K and income that is <=50K come from white people, who take up the majority for the entire dataset.

A graph with text on it

Description automatically generated

1. Feature Selection:

Histogram of numeric feature: comparison between income and numeric feature



A graph of a bar and a graph of a bar

Description automatically generated

A comparison of a graph

Description automatically generated

Correlation of numeric feature:

We did feature selection, which ranks features by how closely correlated they are to the target variable (Income). Absolute value of correlation coefficient.

Figure shows the correlation between the numeric features

A screenshot of a graph

Description automatically generated

1. ***Data Pre-processing***
   1. Encoding

As we found during our initial data analysis, there were few categorical features in the given dataset. There are many machine learning algorithms which can support categorical features in computation without any manipulations but there are many more which do not support. Machine learning algorithm use for this project does not support the categorical feature directly and requires further manipulation in the data. Therefore, we had to figure out how to turn these categorical features into numerical features for algorithm processing.

There are many ways to encode the categorical features into numerical. As with many other aspects of data science, there is no best approach for categorical data encoding. Every approach has its trade-off and potential impact on analysis outcome. Therefore, we tried multiple ways to encode our data and measured its outcome by running all classifiers with cross validation technique.

For our implementation, we used a built-in python encoding package, which provides different techniques to encode the categorical data -example: One-Hot, default dummy encoding. We then used the encoded data to run classifiers with cross validation to measure the encoding technique impact.

Imputation

To handle the missing data, there are two categories of techniques, model based and non-model based approaches. Non-model based techniques include mean imputation and hot-deck imputation. These techniques generally decrease the variance estimates in statistical procedures. Furthermore, these techniques also result in standard errors and bias in results. On the other hand, model based approaches include data mining algorithm techniques to predict the missing values. (For example - Regression model, decision tree, Naive Bayes, etc). This approach results in decreasing the variance as well as bias. For our project, we used three methods to impute the missing data which includes model based and non-model-based techniques.

1. **Mode**

This is the example of a non-model-based approach. According to this approach, we fill the missing feature values with the most frequent data for the respective feature. Hence we are going to replace these null values ‘?’ with the mode. After replacing the null values, the ‘Workclass’, ‘Occupation’, and ‘Native-country’ variable look like this:

* **Workclass**

|  |  |
| --- | --- |
| **BEFORE** | **AFTER** |
|  |  |

* **Occupation**

|  |  |
| --- | --- |
| **BEFORE** | **AFTER** |
|  |  |

* **Native-country**

|  |  |
| --- | --- |
| **BEFORE** | **AFTER** |
|  |  |

* 1. Outliers

We also check if there are any outliers present in the continuous attributes of the dataset. Below are the boxplots showing that outliers are present in our dataset. Hence we are going to remove the outliers by z-score normalization, which we will go into details in the next section.

A screenshot of a computer

Description automatically generated

* 1. Z- Score Normalization

In normalization, giving scores a common standard of zero mean and unity standard deviation facilitates their interpretation.

We use Z-score normalization which replaces the measurement unit with "number of standard deviations" away from the mean.

A screenshot of a computer

Description automatically generated

MinMaxScaller for preprocessing:

A screenshot of a computer

Description automatically generated

* 1. Feature Engineering

For our feature Engineering stage we start with **income** where we see that we have only two values that are to be predicted, either the income is greater than 50K, which is Yes, or the income is less than or equal to 50K, which is No. Hence, we replace ‘<=50K’ and label it as ‘0’ while ‘>50K’ is replaced as ‘1’ and now income is an int data-type.



As for **Education**, 9th-12th grade are recognized as High-School Grad but it is mentioned separately. Hence, we group them back together with HS-grad. We will also create an “Elementary” object for 1st-4th, 5th-6th, and 7th-8th grade.

In terms of **Marital status**, ‘Married-civ-spouse’, ‘Married-spouse-absent’, and ‘Married-AF-spouse’ comes under the category ‘Married’, while ‘Divorced’, ‘Separated’ comes under the category ‘Separated’. Hence, we are grouping them.

Similarly, for **Workclass** ‘Self-emp-not-inc’ and ‘Self-emp-inc’ comes under the category ‘Self-employed. Local-gov, State-gov, and Federal-gov come under the category of government employees.

We use SMOTETomek technique to combines oversampling, imbalanced “income”.

A graph of values distribution

Description automatically generatedA graph of values in a class

Description automatically generated

1. ***Algorithms***

We started with the splitting of the training and testing data, and for that, we checked what the random state was. We choose 5 different algorithms, (logistic regression, decision tree, random forest, KNN, SVM). After that, we checked 5 different models to check which model gives the best accuracy. The best accuracy score is given by the Random Forest Classifier model. To avoid bias and overfitting or underfitting, we cross-validate the models and check the mean accuracy score of them.

Cross-validation: Cross-validating the models to see if they are underfitting, or overfitting and to prevent bias. We will compare the mean accuracy scores of the model.

Hyperparameter tuning: Also, we set parameter values before the learning process begins, it's called Hyperparameter Tuning. We did a Grid search for optimizing hyperparameters.

**Set of hyperparameters:**

* n\_estimators = number of trees in the forest
* min\_sample\_leaf

**n\_estimators :** This is the number of trees we want to build before taking the maximum voting or averages of predictions. Higher number of trees gives better performance but makes code slower. so we choose as high value as your processor can handle because this makes your predictions stronger and more stable.

**min\_sample\_leaf:** Leaf is the end node of a decision tree. A smaller leaf makes the model more prone to capturing noise in train data. Hence it is important to try different values to get a good estimate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model Names | Logistic Regression | Random Forest Classifier | Decision Tree | KNN | SVC |
| Accuracy Score | 71.04890706085862 | 86.49158672747286 | 84.21135398647586 | 79.87891177858154 | 73.83236357917912 |
| Mean Accuracy Score | 70.69256277067836 | 88.5453204075365 | 86.01183139935993 | 81.53902497919141 | 73.47374725705154 |
| Difference between accuracy and mean accuracies | 0.35634429018026026 | -2.0537336800636297 | -1.800477412884078 | -1.6601132006098709 | 0.358616322127574 |

Random forest classifier is the best model with highest cross validating mean score and accuracy score. We will use it for model building.

After the hyperparameter tuning, the best parameters for Random Forest Classifier are 'crietrion' = 'entropy', 'min\_samples\_split' = 2, 'n\_estimators' = 100. We build the model using these parameters.

1. ***Evaluation***

We have built the model after the cross validation and hyper parameter tuning. It is now time to evaluate the model using the classification report, confusion matrix and ROC curve.

After the model evaluation

A graph of different colored lines

Description automatically generated

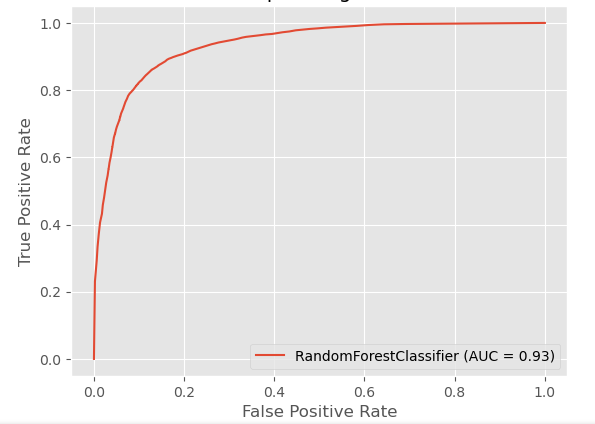
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classifier Name | Precision | Recall | F1-score | Support | Accuracy |
| Random Forest | 0.86 | 0.87 | 0.87 | 6362 | 86.5230382135556 |
| Decision Tree Classifier | 0.84 | 0.85 | 0.84 | 6362 | 84.21135398647586 |
| Logistic Regression classifier | 0.72 | 0.69 | 0.70 | 6362 | 71.04890706085862 |
| SVC | 0.70 | 0.84 | 0.86 | 6362 | 73.83236357917912 |
| KNN | 0.76 | 0.87 | 0.81 | 6362 | 79.87891177858154 |

We get the precision and recall for both the target variable as 0.86 and 0.87. The f1- score of the model is 0.87. The ROC curve gave us the AUC score which is 0.93. Model evaluation gives the results that the prediction is very accurate.

We also checked AUC curve for all other Classifiers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Random forest Classifiers | Logistics Regression | Decision Tree | KNN | SVM |
| AUC | 0.93 | 0.80 | 0.84 | 0.88 | 0.83 |

AUC curve for Random Forest Classifier:



AUC curve for Logistic Regression:

A graph with a red line

Description automatically generated

AUC curve for Decision Tree Classifier:

A graph with a red line

Description automatically generated

AUC curve for K-nearest neighbours classifier:

A graph with a red line

Description automatically generated

AUC curve for SVC Classifier:

A graph with a red line

Description automatically generated

***Conclusion:*** The results obtained above can be used as a standard point of reference for other comparative studies done in the field of predicting values from census data. This comparative study can further be used as a basis for improving the present classifiers and techniques resulting in making better technologies for accurately predicting income level of an individual.

We often hear about “gaps” between gender, race, income, and several other societal factors. Some of these gaps exist for unknown reasons, however, some exist for reasons that are known but not enough has been done to address them. This study serves to objectively identify whether or not income gaps exist based on social factors that should not necessitate an income gap. It does not identify the root cause or problem but rather identifies where further research should be done. It helps to narrow the scope of future research that may identify the problems contributing to the gaps.

Researching these gaps and ultimately their root causes allow us to determine what measures need to be taken to begin closing the gaps. Analyzing the census data from 1994 and comparing it with a more recent census could provide unique insight into whether or not anything has been done to improve incomes gaps related to social factors.

*Python Code Link*

GitHub

*Team Collaboration*

|  |  |
| --- | --- |
| Tasks | Completed By |
| Introduction/motivation to the problem | Mukta |
| problem definition/formalization | All group members |
| Approach | All group members |
| Algorithms | Mukta (Logistic Regression), Jason and Mukta (Decision Tree and Random Forest), Teja (kNN) |
| Results | Mukta (Logistic Regression), Jason and Mukta (Decision Tree and Random Forest), Teja (kNN) |
| Evaluation | Mukta (Logistic Regression), Jason and Mukta (Decision Tree and Random Forest), Teja (kNN) |
| Conclusions and discussion | All group members |