Salary earning prediction depending on different variables - Classification dataset

**Student ID: 9753941**

**Coventry GitHub Repository URL**: < <https://github.coventry.ac.uk/singhh48/9753941-HS-s1> >

Table of Contents

[2 Introduction 3](#_Toc120062943)

[2.1 Problem statement 3](#_Toc120062944)

[2.2 Existing Approaches 3](#_Toc120062945)

[3 Analysis 4](#_Toc120062946)

[4 Pre-Processing 5](#_Toc120062947)

[5 Applying Machine Learning algorithms 5](#_Toc120062948)

[6 Model Tuning 5](#_Toc120062949)

[7 Evaluating the models 5](#_Toc120062950)

[8 Conclusions 5](#_Toc120062951)

[8.1 Comparing the approaches and results of other existing pieces of work 5](#_Toc120062952)

[9 Bibliography 5](#_Toc120062953)

[10 Appendix 7](#_Toc120062954)

Table of Figures

[Figure 1 Salary vs. diff sex counts 4](file:///C:\Users\harma\Documents\6006cem\Coursework\6006CEM_9753941-HS-s1.docx#_Toc120118065)

[Figure 2 Count of salary values with percentage 4](file:///C:\Users\harma\Documents\6006cem\Coursework\6006CEM_9753941-HS-s1.docx#_Toc120118066)

[Figure 3 Marital-status counts 5](file:///C:\Users\harma\Documents\6006cem\Coursework\6006CEM_9753941-HS-s1.docx#_Toc120118067)

[Figure 4 Education level counts 5](file:///C:\Users\harma\Documents\6006cem\Coursework\6006CEM_9753941-HS-s1.docx#_Toc120118068)

[Figure 5 Work class counts 5](file:///C:\Users\harma\Documents\6006cem\Coursework\6006CEM_9753941-HS-s1.docx#_Toc120118069)

# Introduction

Predicting the salary of an individual can be a challenging task, however with the help of Machine Learning and computing power, it’s possible to train the machine in such a manner that when we pass it different categorical and numerical values, we can predict if and individual will earn less of more than $50.000 dollars per year, also with the use of Machine learning techniques its possible to identify if some immutable characteristics that differentiate an individual from another can also be impactful in their respective ability to earn below or above the $50.000 dollar threshold. In this report, the dataset will meticulously be analysed and prepared to be processed through different Machine Learning algorithm to achieve the highest prediction accuracy possible.

## Problem statement

The dataset selected and used throughout in this report has been sourced from kaggle.com/datasets/ayessa/salary-prediction-classification and it contains the 1994 US income census data of working Adults. There are 32.6K instances in total, all containing different variables and values which can be a factor in determining the annual income of certain population demographic. Predicting the salary of an induvial does not necessarily mean that is what they will earn during their career; however, it can be used as a method for choosing a career path that can lead to being part of the desired income bracket.

## Existing Approaches

During the research and literature review for this dataset, I have discovered that the most common approach used by other users on Kaggle.com has been Logistic Regression, followed by K-Nearest Neighbours and Random Forest Classifier, in average the accuracy score for Logistic Regression on multiple notebooks has been reported ranging from 80% to 85%, I will be using this figure numbers as a reference to what the accuracy score should be most close to, and to try and use other algorithms such Neural Networks and SVM to get higher score where possible.

# Analysis

The dataset selected, has fifteen columns and 32561 instances of data, however, I may have to reduce the number of columns to only select the data that is more relevant to the kind of prediction I am looking to achieve. The following table contains all the names of the columns and what they represent.

The first 14 columns are made of all the attributes that may correlate with how much an individual is able to earn each year, however, not all data present is composed of numerical values, some of which are categorical, meaning that they will need to be properly processed and encoded in number before being passed to the machine learning algorithm

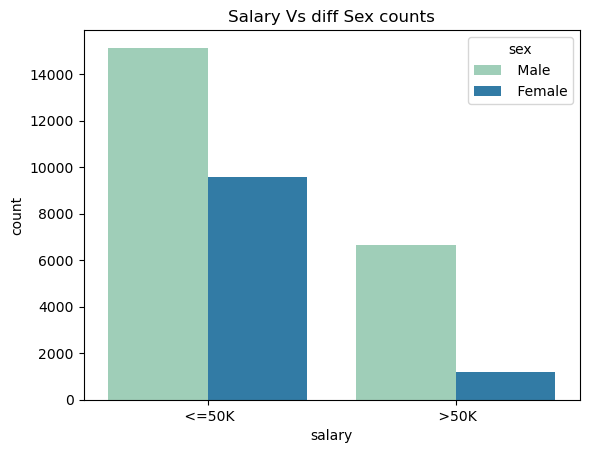
Table . Names of the Attributes

|  |  |
| --- | --- |
| Attributes | |
| Attribute name: | Description: |
| Age | Int value > 0: individual’s age |
| Work class | Categorical: Employment status of the individual |
| Fnlwgt | Final weight number int value > 0: count of individuals the census instance represents |
| Education | Categorical: Highest education of the individual |
| Education-num | Int value > 0: Education level enumerated |
| Marital-status | Categorical: Specifies the marital status of the individual |
| Occupation | Categorical: Specific occupation of the individual |
| Relationship | Categorical: Relationship relative to others |
| Race | Categorical: Individual’s race |
| Sex | Categorical: Individual’s biological sex |
| Capital-gain | Int value > 0: Income made from investing |
| Capital-loss | Int value > 0: Income lost from investing |
| Hours-per-week | Int value > 0: Individual’s reported weekly hours |
| Native-country | Categorical: individual’s country of origin |
| Salary | Prediction value greater, less, or equal to 50.000 dollars |

During the analysis phase of the dataset, I found out that, although there appear to be no Null or NaN values present, 3 columns out of 15 have “?” values, indicating that the data might be missing or not complete, possibly affecting the accuracy of the machine learning algorithm if used in this state with no changes, to find the missing values I used functions to print out each unique value contained in each column, identifying where the missing values are, and finally from the said column I took the total count of the missing values before replacing them for NaN, which makes them easier to handle with the pre-existing sklearn functions.

## Data visualization

Figures down below, (*Figure 1)* is the total count of salary attribute values, shown in a bar chart form with their respective percentage value on top, and finally, (*Figure 2)* shows how much the two different sexes make each year below or above the $50.000 threshold.

Chart, bar chart

Description automatically generated

Figure Salary vs. diff sex counts

Figure Count of salary values with percentage

Chart, bar chart

Description automatically generatedShowing the total number of values count is helpful to understand if the data is more skewed to one side than the other and if there are a small number of instances that require to be oversampled or under sampled, in the case of the dataset used there is no need for oversampling of the >50K data as the total ratio is 24:100 (*(Chingis), 2021*) and oversampling is usually recommended on dataset sample with ratios of 1:100 for the minority of the values, I have not performed any oversampling on the dataset in order to not increase the chances of overfitting the data, as oversampling would create more of the instances containing >50K, by duplicating them randomly.

During data visualization I have found out more discrepancies in the data that I will have to fix during pre-processing such as:

Chart

Description automatically generatedThe ‘marital-status’ attribute has three values that means ‘married’ however they have been divided in three categories, married civ spouse and AF spouse, respectively meaning civilian and Armed forces, and finally Married-spouse-absent, can be categorised simply as married to reduce the number of categorical values that have small counts. (*Figure 3*)

Figure Marital-status counts

In the ‘Education’ attribute I have observed that from preschool to grade 12th, the categorical data can simply be summarized as ‘school’, instead of having eight different values. (*Figure 4*)

Chart

Description automatically generatedFinally in the ‘workclass’ attribute I have found out that there are Without-pay and Never-worked values, that can be considered as outlier in a dataset that is used to predict the yearly salary of an individual, during pre-processing I will be eliminating all possible outlier that will have an effect on the accuracy of the algorithm. (*Figure 5*)

Figure Education level counts

Figure Work class counts

# Pre-Processing

In this phase, I need to prepare the data and make sure all re-adjustment are made from the finding done in Data analysis and visualization, therefore, I have replaced all missing data with NaN and removed them, reducing the today instances from 32561 down to 30139, value ‘Never-worked’ of work class attribute was part in the instances removed, however ‘Without-pay’ had to be removed separately, I have combined the small value in marital-status and education attribute to one simple category, married and school respectively, and finally I have completely removed the attribute that would not be fit for purpose or were not of any meaningful importance such as ‘fnlwgt’, ‘capital-gains/loss’, and ‘education-num’ as ‘education’ represents the same values but that are not yet been encoded in numerical digits.

StandardScaler removes the mean and scales the data to unit variance. However, the outliers have an influence when computing the empirical mean and standard deviation which shrink the range of the feature values as shown in the left figure below. Note in particular that because the outliers on each feature have different magnitudes, the spread of the transformed data on each feature is very different: most of the data lie in the [-2, 4] range for the transformed median income feature while the same data is squeezed in the smaller [-0.2, 0.2] range for the transformed number of households.

StandardScaler therefore cannot guarantee balanced feature scales in the presence of outliers.

MinMaxScaler rescales the data set such that all feature values are in the range [0, 1] as shown in the right panel below. However, this scaling compress all inliers in the narrow range [0, 0.005] for the transformed number of households.

# Applying Machine Learning algorithms

# Model Tuning

# Evaluating the models

# Conclusions

## Comparing the approaches and results of other existing pieces of work

# Bibliography

(Chingis), C. (2021, February 6). *Introduction to synthetic minority over-sampling technique and its implementation from scratch*. Medium. Retrieved November 23, 2022, from https://towardsdatascience.com/introduction-to-synthetic-minority-over-sampling-technique-and-its-implementation-from-scratch-77593647c10d

Chet Lemon, Chris Zelazo, & Kesav Mulakaluri. (2018). Predicting if income exceeds $50,000 per year based on 1994 US Census Data with Simple Classification Techniques. In *https://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf*. UC San Diego. <https://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf>

Garg, R. (2021, October 7). *7 Types of Classification Algorithms*. Analytics India Magazine. <https://analyticsindiamag.com/7-types-classification-algorithms/>

*Machine Learning Classification – 8 Algorithms for Data Science Aspirants*. (n.d.). Https://Data-flair.Training/. Retrieved 16 November 2022, from <https://data-flair.training/blogs/machine-learning-classification-algorithms/>

*Salary Prediction Classification*. (2022, May 21). Kaggle. https://www.kaggle.com/code/mahmoudftolba/salary-prediction-classification

TOLBA, M. (2022, May 21). *Salary Prediction Classification*. Kaggle. <https://www.kaggle.com/code/mahmoudftolba/salary-prediction-classification>

R JAIN, T. (2022, August 18). *Feature Selection Methods*. Kaggle. <https://www.kaggle.com/code/tarundalal/feature-selection-methods/notebook>

ALI OMAR, A. (2022, October 14). *Salary Prediction Classification | 88% Accuracy*. Kaggle. <https://www.kaggle.com/code/ahmedaliomar/salary-prediction-classification-88-accuracy>

HAMDY, A. (2022, May 9). *Salary Predection 86.40% Accuracy*. Kaggle. <https://www.kaggle.com/code/abdo977/salary-predection-86-40-accuracy>

*Difference between Standard scaler and MinMaxScaler*. (2018, July 9). Stack Overflow. <https://stackoverflow.com/questions/51237635/difference-between-standard-scaler-and-minmaxscale>

*seaborn.kdeplot — seaborn 0.12.1 documentation*. (n.d.). Retrieved 15 November 2022, from <https://seaborn.pydata.org/generated/seaborn.kdeplot.html>

*User guide: contents*. (n.d.). Scikit-learn. Retrieved 16 November 2022, from <https://scikit-learn.org/stable/user_guide.html>

*What is a good range of values for the svm.SVC() hyperparameters to be explored via GridSearchCV()?* (2014, October 13). Stack Overflow. <https://stackoverflow.com/questions/26337403/what-is-a-good-range-of-values-for-the-svm-svc-hyperparameters-to-be-explored>

# Appendix

< A suggested checklist for you, for full details please refer to the coursework brief >

1. The following naming convention is used for the Coventry GitHub Repository and Coventry OneDrive

StudentID-Initials-s1

For example, a student Liz Truss whose student ID is 12345678 would name their repository or shared folder as **12345678-LT-s1**

Failing to follow the naming convention may delay the release of marks and feedback for your coursework.

1. **Coventry** GitHub Repository URL **or** **Coventry** OneDrive URL: added to the top of this report
   1. Coventry GitHub Repository includes:

* URL of the selected dataset(s) included in README
* The original selected dataset(s)
* Source-code (.ipynb)
* Demonstration video (.mp4)

1. Source-code added **as text** in Appendix B (below)
2. Submission in the form of a **Word** document. *\*\*Other format is not accepted.*

|  |
| --- |
| **Appendix B** |

< **Replace** this instruction with all the Programming Code for the coursework.

Make sure you have highlighted and referenced any code not written by you >

< **DO NOT** use screenshots of your code here. Your code should be presented **as text**.

There are many good tools to help you format your code such as <http://hilite.me> >

< You can select and copy **all code at once** in a notebook by:

1. Graphical user interface, text, application

   Description automatically generatedclicking in any cell of the notebook, the cell will be highlighted in green as below
2. Graphical user interface, text, application

   Description automatically generatedthen press Esc on your keyboard, the selected cell will be highlighted in blue as below
3. now you can Ctrl+A to select all cells of the notebook

Graphical user interface, text, application

Description automatically generated

1. and then copy and paste as normal to some tool, such as hilite.me above, make sure you select a correct language (Python), then click Hightlight

Graphical user interface, text, application

Description automatically generated

1. you now can select the text in the Preview and copy and paste it over to this Appendix

Graphical user interface, text, application, email

Description automatically generated

1. finally, remember to remove all text in this instruction for this Appendix >