Machine Learning report

Jorge Vicente Puig

Summary

Machine Learning models of a data set containing wages of US male workers from 1988 has been done among this report. It has been created a regression model for predicting the wages and a classification model, in order to extract social conclusions as race pay gap.

UAB
Universitat Autònoma
de Barcelona

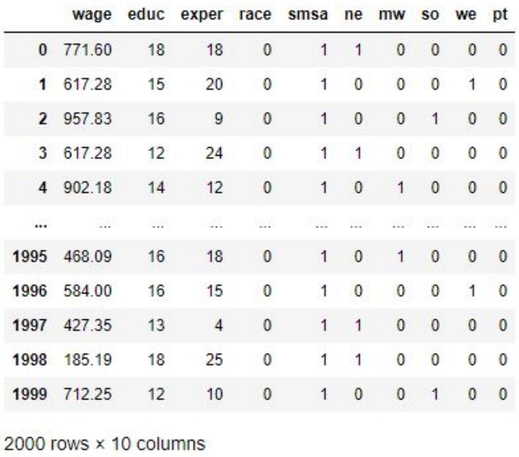
December 2021

Contents

1 Introduction

In this report is explained the analysis of the data set *uswages* from *faraway* package. The data set contains the following information:

- **wage:** wage of the worker.
- **educ:** years of education.
- **exper:** years of experience.
- **race:** 1 if Black, 0 if white (other races not in sample).
- **smsa:** 1 if living in Standard Metropolitan Statistical Area, 0 if not.
- **ne, mw, we, so:** 1 if living in the North East, Midwest, West or South respectively.
- **pt:** 1 if working part time, 0 if not.



	wage	educ	exper	race	smsa	ne	mw	so	we	pt
0	771.60	18	18	0	1	1	0	0	0	0
1	617.28	15	20	0	1	0	0	0	1	0
2	957.83	16	9	0	1	0	0	1	0	0
3	617.28	12	24	0	1	1	0	0	0	0
4	902.18	14	12	0	1	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
1995	468.09	16	18	0	1	0	1	0	0	0
1996	584.00	16	15	0	1	0	0	0	1	0
1997	427.35	13	4	0	1	1	0	0	0	0
1998	185.19	18	25	0	1	1	0	0	0	0
1999	712.25	12	10	0	1	0	0	1	0	0

2000 rows × 10 columns

Several questions arise from the previous dataframe: Is there a wage gap due to race? Could I make a good prediction of worker's wage?

For solving this questions , it has been done a construction of two Machine Learning models[1]:

- A regression model for predicting the wage.
- A classification model for classifying workers by race.

[1] **Remark:** All code is written in *Python* using *Jupyter Notebooks*. It can be found at the following *GitHub* repository: https://github.com/JorgeVicentePuig/Master/tree/main/Machine_learning or in the attachment.

2 Strategy

In order to implement the proposed Machine Learning models the following general strategy has been used:

1. Import *uswages* data set from *faraway* package.
2. Preprocessing of the data analysing the features, removing the outliers and splitting it.
3. Different Machine Learning models have been tested tunning the hyperparameters.
4. An analysis of performance of the models with proper metrics.
5. Explanation of the selected model.

The general strategy is summarized in the following image,

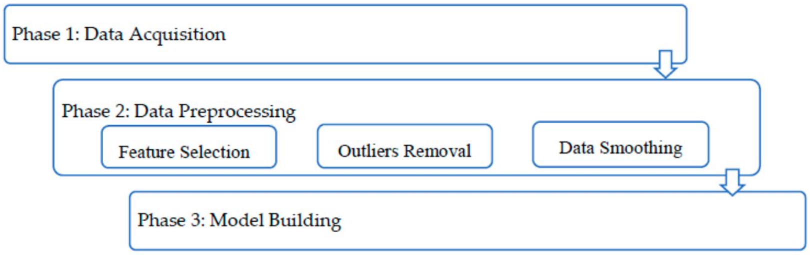


Figure 1: General strategy

◆ **Error measurement metrics**:

- For the regression models it have been used the next metrics:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \qquad \text{MSE} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \qquad R^2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

- For the classification models it have been decided to use the confusion matrix *(also the normalized confusion matrix)*.

3 Procedure

3.1 Regression models

For the wage prediction it has been decided to implement the models: **Random Results**, **Naive Mean**, **Linear Regression**, **Random Forest** and **Multilayer Perceptrons**, i.e. forward neural network.

In the case of Multilayer Perceptrons it has been done an implementation without selecting the hyperparameters and another one with the hyperparameters that reduce the Root Mean Squared Error, i.e.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}(y_i - \hat{y}_i)^2}$$

Configuration of the hyperparameters for the wage-predicting model[2]

Parameter	**Value**
Activation function	Relu
Solver option	Lbfgs
Hidden layers	(12)
Max iterations	500
Alpha	0.00010
Randomness	10

The performance of the models with the selected metrics is contained in the following table:

Metrics for the different models

Models	**MSE**	**MAE**	R^2
Random	372222.768314	505.935892	-2.994423
Naive mean	94092.383194	251.177009	-0.009731
Linear Regression	55289.827759	186.700028	0.40667
MLP	53902.034556	184.341479	0.421563
Tunned MLP	49931.308855	177.126819	0.464174
Random Forest	54349.601022	186.006658	0.416760

Looking at the results the best model to use for the prediction is the **Tunned MLP**. We will explain this model.

In order to make an analysis of the Neural Network the use of a specialised library is required. In fact, neural network are *"black box"* algorithms and then *sklearn* can't provide information.

The library **SHAP (SHapley Additive exPlanations)** is based on a game theory approach and is able to explain the output of any machine learning model. Therefore, by using this library we obtain the next bar plot,

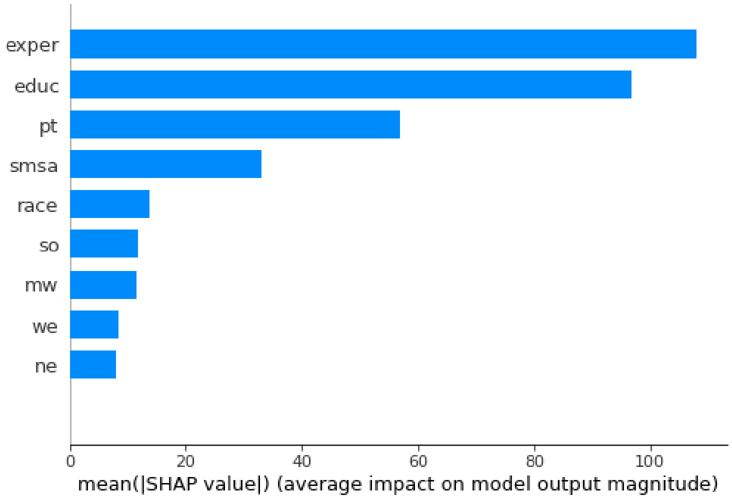


Figure 2: Explanations obtained with SHAP

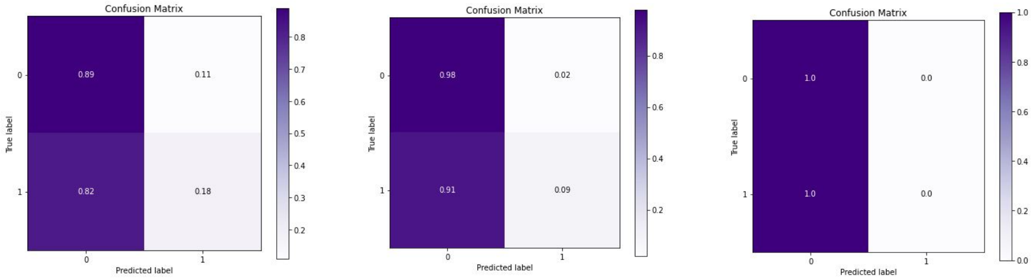
Now, we are able to obtain a deeply idea of what the model does. In fact, we can see that the more important features are experience, education and part time.

[2] The code and a more detailed explanation can be found on the Regression Model *Jupyter Notebook*.

3.2 Classification models

For the classification task, it has been decided to test the models: **Decision Tree**, **Random Forest** and **Logistic Regression**. The objective is to make a binary classification of the workers by race (1 if black, 0 if white).

Following the strategy explained in *Section 2* it has been preprocessed the data, built the classification models and measured the errors. The results are presented in the following confusion matrices:



(a) Decision Tree (b) Random Forest (c) Logistic Regression

Figure 3: Confusion matrices for the different models

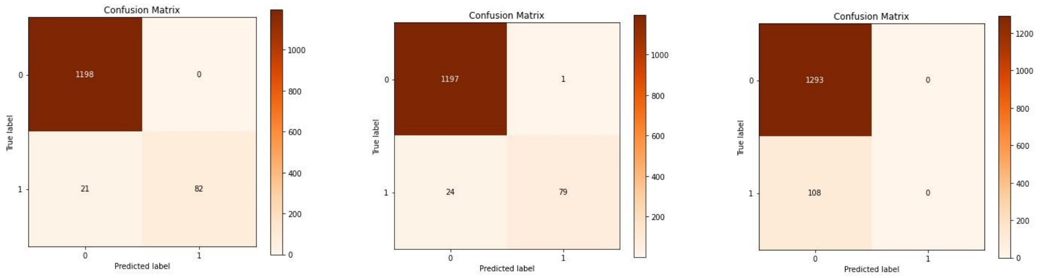
As can be seen, all models do the same mistake: **always tend to predict that the worker is white**.

The reason of this mistake come from the data. Indeed, the data set *uswages* is really imbalanced on the ratio of white workers vs black workers: we only have 155 black workers whereas there are 1777 white workers.

The strategy used for solving this problem is called a **Oversampling technique**: we will split the data on training set and test set, then on training set we will increase the number of black workers by duplicating them randomly[3].

Therefore, using this technique the data set will contain $1.3 * 155 \approx 201$ black workers. Notice that, even the data set is still imbalance, with this slightly change we obtain the following results,

[3] A more detailed explanation along with the code is on the Classification Model *Jupyter Notebook*



(a) Decision tree with balance data

(b) Random Forest with balance data

(c) Logistic Regression with balance data

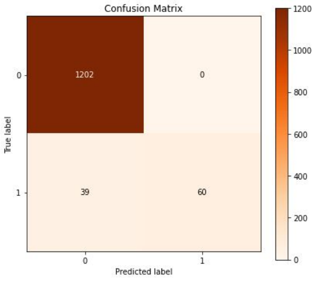
Figure 4: Models with balance data

Firstly, notice that for *Logistic Regression* no improvement has been achieved. However, for *Decision Tree* and *Random Forest* models it could be avoided the mistake. Now we obtain a more balance prediction, still not perfect, but is more acceptable.

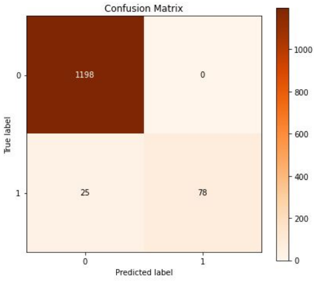
On the Oversampling technique it has been decided to balance data by 1.3 times. Then, the question arise naturally, is the parameter 1.3 the best one possible?

In order to give an answer, it has been done a discretisation of the real interval $[1, 10]$ by an step of 0.5, i.e. from data without sampling to equal number of black and white people.

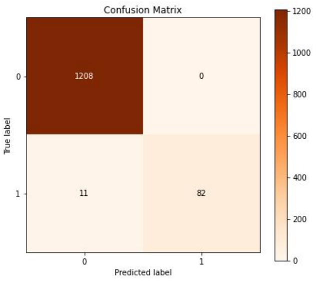
It has been decided to use the *Decision Tree model with depth 25*, due to the good performance obtained, to test the performance.



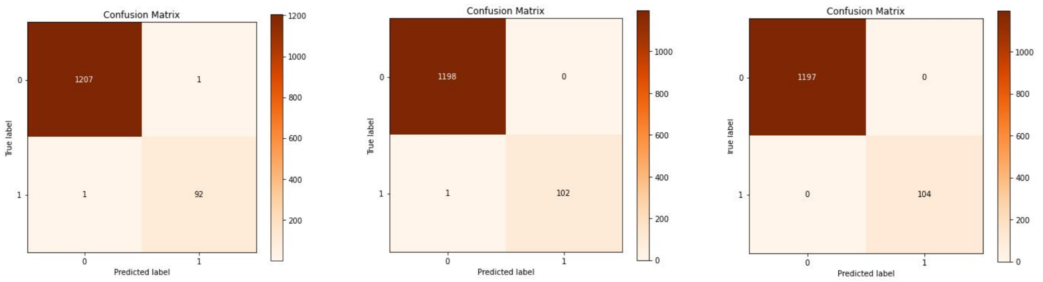
Without balance data



Balance data by 1.5



Balance data by 2



Balance data by 4

Balance data by 5

Balance data by 9

Confusion matrices for different data

As we can see, the results are highly improved when increasing the sample data of black workers. Even, for samples bigger than 4 we almost obtain a perfect classification model.

In order to get a deeply understanding of the model, it has been computed the feature importance,

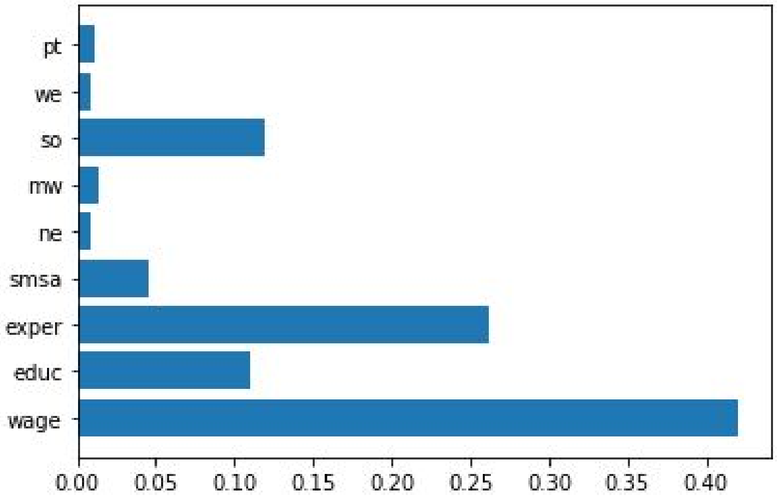


Figure 7: Feature importance on classification model

We can see that the more relevant features are wage, experience and south.

4 Conclusion

Regarding to **regression model** we can extract two main results:

- We can obtain better results by increasing complexity model, however on this report we weren't able to make a close prediction of the wages. By tunning the neural network, we have obtained the best model however it is more computationally expensive than other models, as the linear model, which a quick implementation give us good results.

- By using the *SHAP* library we were able to analyse and have a deeper understanding of the machine learning models, which is essential in a project.
  In our case the experience, education and part time were the important features. Moreover, is interesting to notice that living or not in the Standard Metropolitan Statistical Area is the fourth importance feature.

Regarding to **classification model** we can extract three main results:

- Imbalance data can imply bias and incorrect classification models. Consequently, applying oversampling/undersampling techniques can be crucial to obtain good and accurate classification models.

- Using an oversampling technique lead us to almost perfect results, which could let us think that there is a race gap.

- By looking at the feature importance we see that wage, experience, education and south are the more relevant features. Is interesting to compare the importance that variable south has compared with north east, midwest and west.

References

[1] Andreas C. Müller and Sarah Guido *Introduction to Machine Learning with Python* https://github.com/amueller/introduction_to_ml_with_python

[2] Llorenç Badiella Busquets, Joan Valls Marsal *Course: Classification and Regression Trees and Neural Networks with R*

[3] *Evaluation of Different Deep-Learning Models for the Prediction of a Ship's Propulsion Power* https://www.mdpi.com/2077-1312/9/2/116