MISCADA Core IIA (3) Classification

Code GitHub URL: https://github.com/lunamox11/MISCADA_CoreIIA_Classification

Executive Summary

The adult data set includes 9 factor variables (workclass, education, marital_status, occupation, relationship, race, sex, native_country and income) and 5 numeric variables (age, education_num, hours_per_week). As required to do the classification model, so I choose one of data factors income as the data task. Annual income is the target feature and is measured as a binary category target. The income is divided into two classes, one is >50K, the other is <=50K. I need to build a model according to the exist data to predict the income class of a new person depends on his workclass, education, marital_status, occupation, relationship, race, sex, native_country, age, education_num, hours per week and so on. In other word, the classification model will be used to select candidates for a new service offered by the sponsor of the project targeting individuals with salaries exceeding fifty thousand US dollars.
To make the prediction more correctly, I need to try lots of models to find out which model performs best which means it has the greatest probability to predict a new person's income. The data set is just a small sample of the American people, if the model can fit the sample data well, it can also give us a general overview of the income of American people. Even we can find out which factors affect the income most in America.

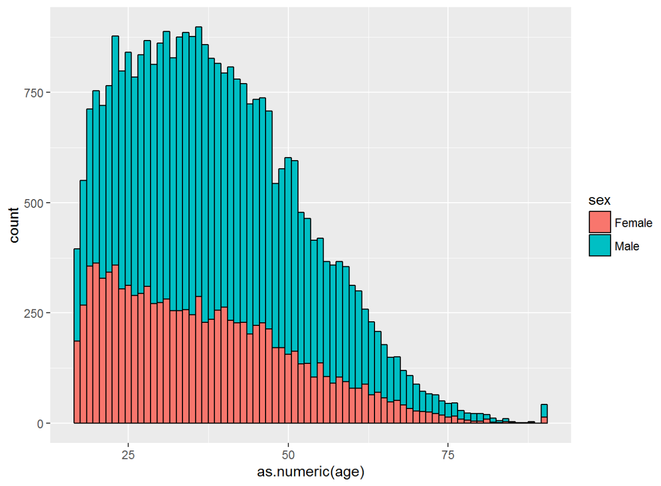

Figure.1 age count

However, I find the sex sample is imbalanced (in figure.1). According to the graph below, it is apparently females are underrepresented. The data set mostly represent the male group rather than the whole two gender groups.

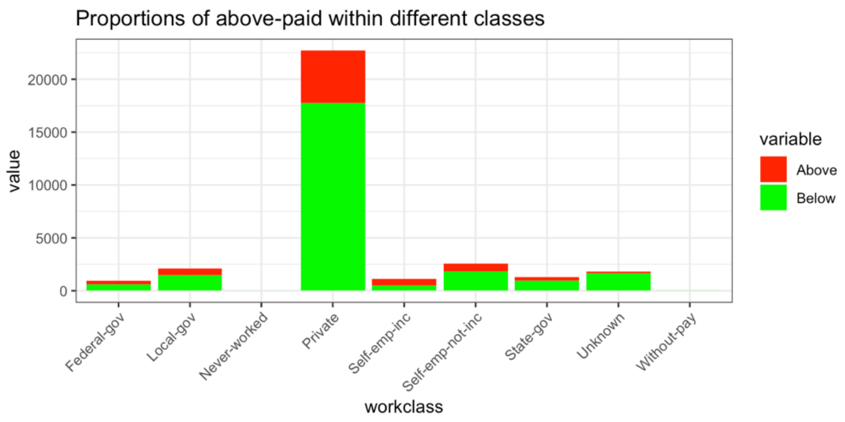

Figure.2 Work class

From figure 2, it is noticed that the private sector, which has the most people work in, has the largest number of population that earn more than 50K per year. However, in terms of the proportion, the self-employed people are the winner.

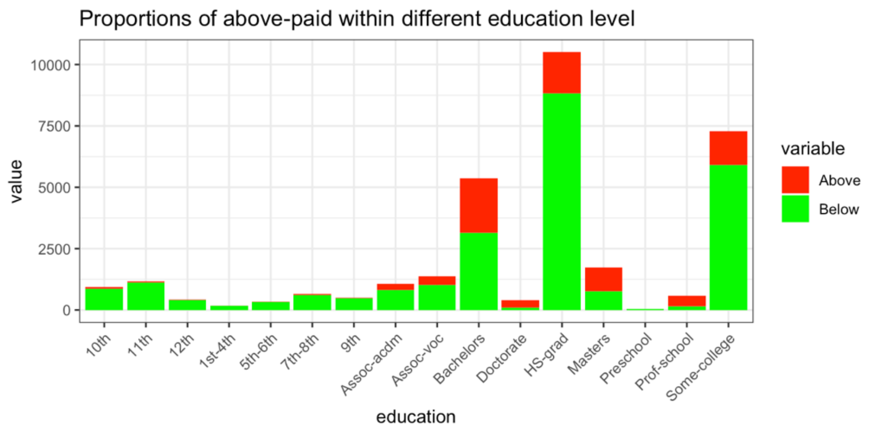

Figure.3 Education

From figure 3, It is worth noting that the level of higher education may lead to higher possibilities for higher paid jobs. But this must be a difficult process, because most people cannot complete it.

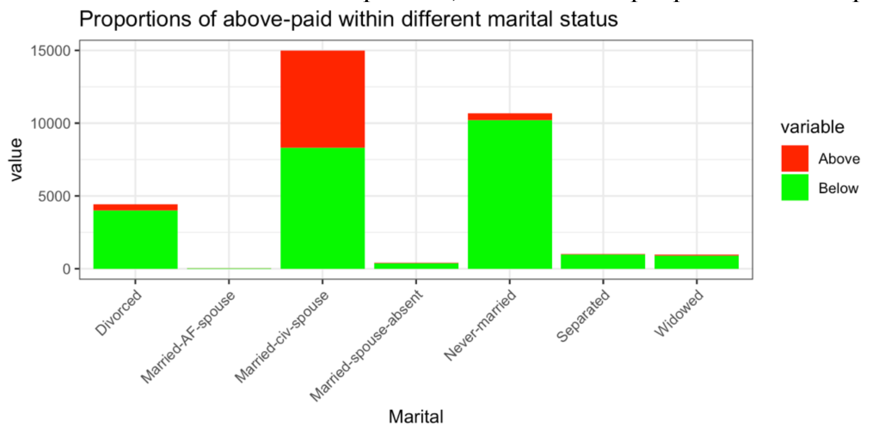

Figure.4 Marital

From figure 4, it is noticed that a good relationship is also very important for earning more money. Misclassification is the thing we should avoid in modelling. In other word, the model should not put the new person into the wrong income class. Person who earns more 50 thousand but in the group whose income less than 50 thousand or person who gains less than 50 thousand but in the group whose income more than 50 thousand will received unfair treatment.

I use serval models, such as random reforest, support vector machine, neural network and so on. Then I compare the accuracies of each model to find out which performs better. Neural network is to simulate lots of densely interconnected brain cells so you can get it to learn things and make decisions like a human.

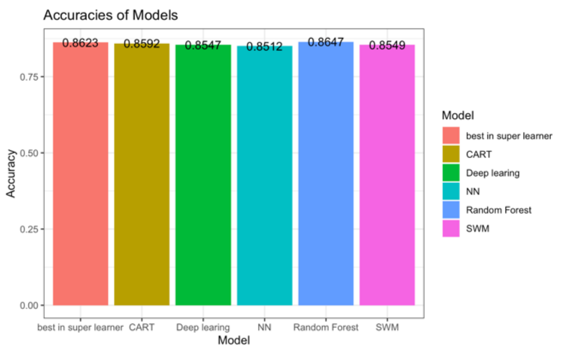

Figure.5 Accuracy

From figure.5, it shows the accuracy comparison between models. The random reforest performs the best in accuracy.

Technical Summary

Exploratory Dataset

The adult dataset has 32,561entries. Each entry contains the following information about an individual:

Table 1. Dataset Description

	Variable	Values						Missing
Polynomials	Workclass	Private (68%), Self employed 1 (8%), Local Gov(6%), State Gov(4%), Unknown (5%), Self employed 2 (3%), Federal Gov(3%), No Pay(1.5%),Never Worked (0.5%)						1836
	Education	High School (32%), Some college (22%), Bachelors (16%), Masters (5%),Vocational (4%), 11th (4%), Assoc Academic (3%), 10th (3%), 7-8th (2%),Professional School (2%), 9th (2%), 12th (2%), Doctorate (1%), 5-6th (1%),1-4th (1%), Preschool (1%)						0
	Relationship	Husband (41%), Not-in-family (26%), Own child (16%), Unmarried (11%), Wife (4%), Other relative (2%)						0
	Race	White (85%). Black (10%). Asian / Pacific Islander (3%), American Indian / Eskimo (1%), Other (1%)						0
	Marital Status	Married-civ-spouse (46%). Never-married (33%). Divorced (14%) Separated (3%), Widowed (2%), Married-AF-spouse (1%), Married-spouse-absent (1%)						0
	Occupation	15 categories						1843
	Country	42 categories: USA (90%)						583
Binomials	Income (Label)	<=$50K (76%), >$50K (24%)						0
	Gender	Male (67%), Female (33%)						0
Real		Mean	Median	Std Dev	Skewness	Kurtosis	Range	
	Age	38.58	37	13.64	0.56	2.83	17 - 90	0
	hours_per_ week	40.44	40	12.35	0.23	5.92	1- 99	0
	education_ num	10.08	10	2.57	-0.31	3.62	1-16	0
	Captial Gain	1078	0	7385	11.95	157.77	0 - 99999	0
	Capital Loss	87.3	0	403	4.59	23.37	0 - 4356	0
	fnlwgt	189778	178356	105550	1.45	9.22	12285 - 1484705	0

From table 1, the high kurtosis in the capital gain and capital loss (157.77 and 23.37 respectively) indicate a long tail. From the boxplots in Figure.6, it is obvious that the range of numeric variable values for age, education number and hours_per_week is slightly higher in >50K class, indicating that these variables may have predictive significance.

Table 2. Income Percentage

Income	Number	percentage
<=50K	24720	75.91904
>50K	7841	24.08096

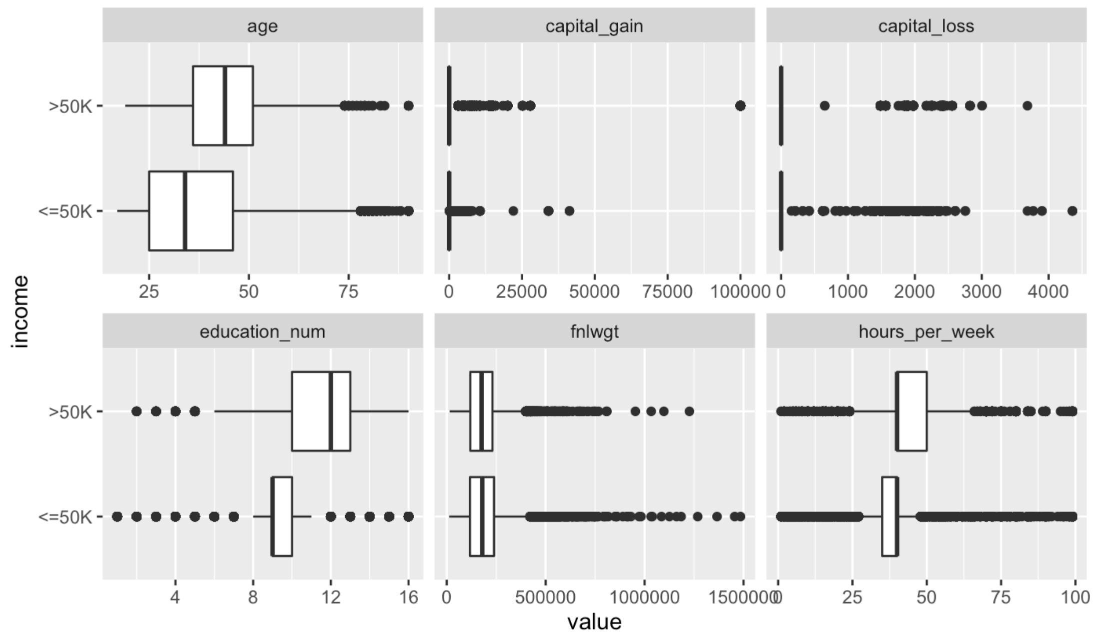

Figure.6 Boxplots of numeric variables

From table.2, it is noticed that the adult original dataset contains a distribution of 24.08% entries labelled with >50k, numbering 7841 and 75.92% entries labelled with <=50k, numbering 24720. It should be similar to the training and test sets which spited from the adult dataset.

Pre-process Data

I tried two strategies, one is cross-validation and nested resampling, the other is splitting train/test/validate set and resampling.

The first strategy:

1. Resampling and fixed cross-validation folds

   I first create the resampling strategy using the rsmp function which allows every 5 cross-validation folds can be chosen. Then 'instantiate' it on the task, after choosing the 5 folds. Instantiating means that the folds get fixed, then the same set of folds can be used on multiple learners. Therefore, comparisons between different models are fair. I use the function `po(learner_cv)` and pipeline to enable the cross validation.

2. Handling Missing Values
   The dataset contains lots of missing values for three variables, workclass, occupation and native_country, which can be dealt with some algorithmic transformations. The missing values are set by a default marker called '?'. I create a pipeline of missing fixes to impute missing values rather than use na.omit() function to deal with the missing values.

3. Encode factor
   The encode pipeline can do one-hot encoding of factors. Some learner models don't accept factors, so I create a pipeline operation to encode them before passing to the learner. I use the function `po(encode)` adds operations and `%>>%` connects the steps. One-hot encoding involves splitting of different factor features into its own category and every category should be binary value. However, the sex factor will not be encoded as it only has two categories, female and male. To avoid of dimensionality, no One-Hot Encoding is done for sex attribute.

4. Fitting
   I first define a super learner as it can combine all models and select which model performs good. Using pipeline to combine all the learners and all the operation altogether. I use classification and regression trees with complexity parameter and without complexity parameter, baseline classifier, random forest and eXtreme Gradient Boosting.

5. Result

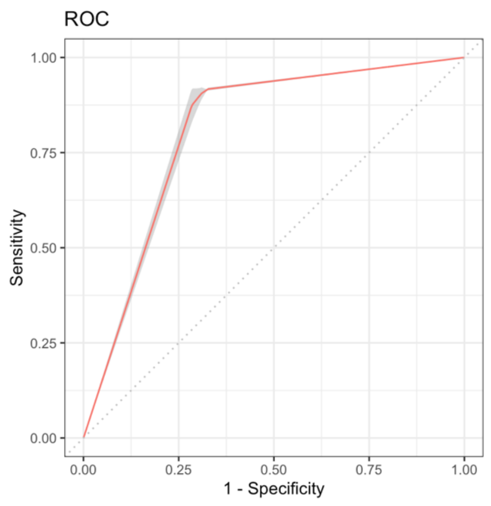

Figure.7 ROC plot of 5CV best learner

These are the best results achieved of all the learners (except in false positive). From the metrics (in table.3), ce(misclassification error), accuracy, AUC, false positive rate, false negative rate. Since I do another strategy later, the metrics will help me to compare different strategy and find out which is best.

Table 3. 5CV metrics

classif.ce	classif.acc	classif.auc	classif.fpr	classif.fnr
0.13768005	0.86231995	0.81286007	0.08673139	0.29830379

The second strategy:

1. Resampling and splitting
   Since I am going to do the deep learning in this strategy, I don't do cross validation here because deep learning is the most computationally expensive method and it needs lots of time to wait. I split the dataset into train set and test set using the resampling function.

2. Missing values
   I use the recipies package and use the pipeline to combine all the operations. As for missing values, I impute missing values on numeric values with the mean, which is better than change it into NAs because it will be closer to the original dataset. And I create a new factor called "unknown" to account for NAs in factors.

3. One-hot encode
   I use step_dummy function to turn all the factors into a one-hot coding, which split different factor features into its own category. Every category should be binary value before it pass to the model.

4. Fitting
   I use the deep learning, random forest, SVM, classification and regression tree and simple neural network. Then I get the probability predictions and raw classes on the test set, which

help to compare these models. I calculate the loss of the train and validation set in the deep learning model.

5. Result

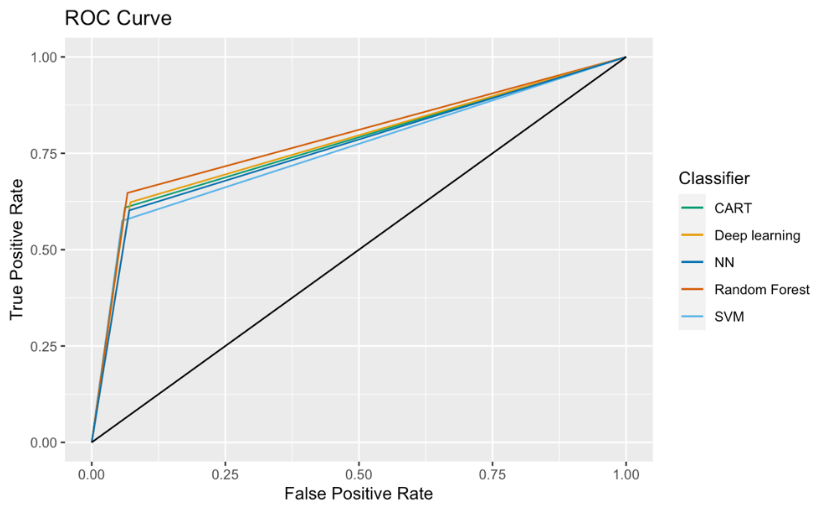

Figure.8 ROC

ROC curve is not smooth enough because model can only provide discrete predictions, rather than a continuous score.

Table.4 Metrics comparison

Model	ACC	AUC
SVM	0.8549	0.8980
CART	0.8592	0.8859
Random Forest	0.8647	0.9095
Deep learning	0. 8572	0.9058
NN	0.8512	0.9064
Best in super learner	0.8623	0.8128

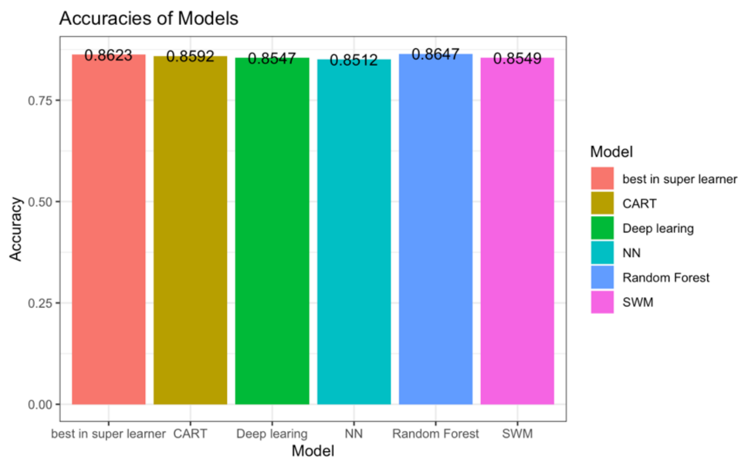

Figure.9 ACC

From the table.4 and figure.5, it shows the comparison between models through AUC and accuracy. The random reforest performs the best in accuracy.