Dataiku Machine Learning Part

Error

This dataset is extremely unbalanced. We can classify the data with an about 94% of accuracy just by classifying every individuals in the class -50K. I thus decided to forget about the classic accuracy measure and evaluate all my models with the help of the Balanced Error Rate. The real challenge with these kind of data is to success in predicting with accuracy the minority class. The Balanced Error Rate (BER) is the average of the proportion of wrong classifications in each class(you can read more about it here http://research.ics.aalto.fi/events/eyechallenge2005/evaluation.shtml).

Random Forest

I decided to train a Random Forest. It is pretty fast to train and give good results when it is correctly used. I first try to learn a classifier without tuning. The results were pretty bad (~ 30% of BER). Indeed the first class was very well classified but the classification of the +50K was awful, I achieved a ~95% of standard accuracy but again this is meaningless since the classification +50K individuals is very bad (almost 60% of misclassification !). This Random Forest is biased by the enormous proportion of the -50K individuals !

Taking weight into account

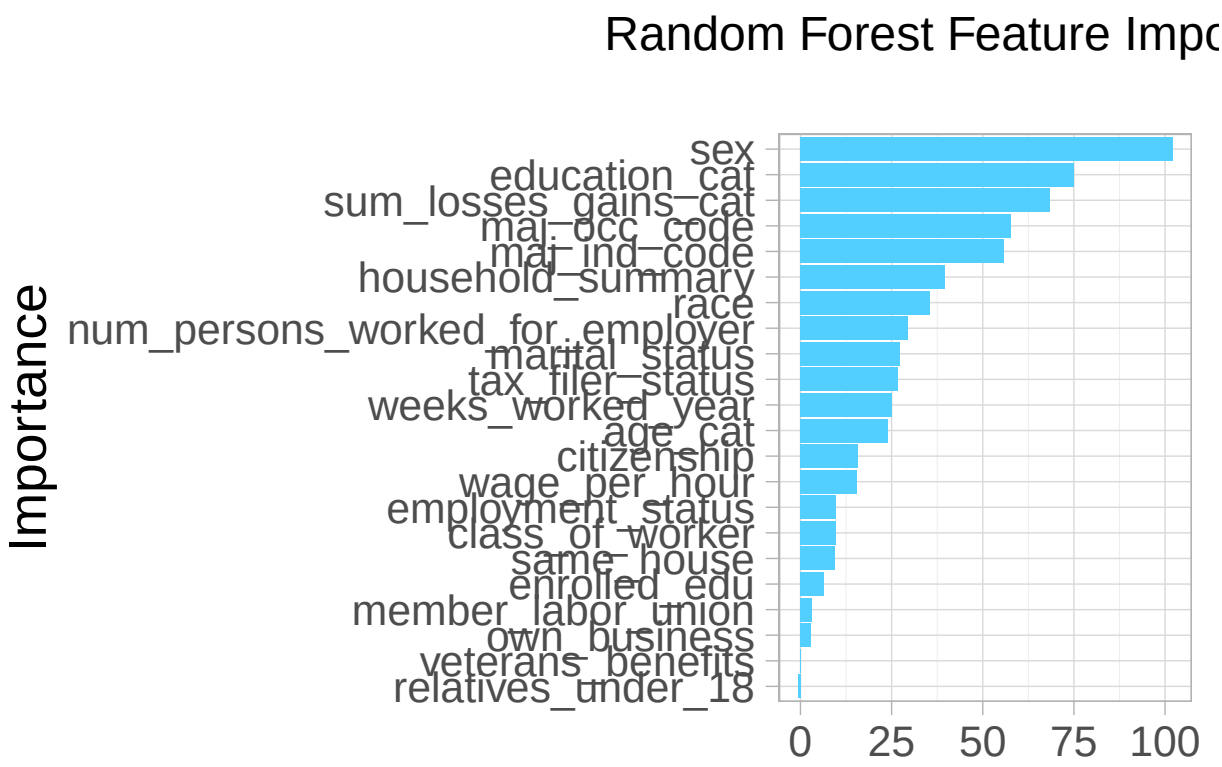
The random Forest library in R let us take into account the weight of each class with the sampsize parameters, it specifies the number of size sample to draw for each class. I first run some RF to get the importance of the features. The graph shows the utility of the features to classify the data

```
imp <- importance(rf, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

p <- ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +
  geom_bar(stat="identity", fill="#53cfff") +
  coord_flip() +
  theme_light(base_size=20) +
  xlab("Importance") +
  ylab("") +
  ggtitle("Random Forest Feature Importance\n") +
  theme(plot.title=element_text(size=18))

p
```

```
## Warning: Stacking not well defined when ymin != 0
```



I removed some useless features that don't help the Random Forest to classify and run another another RF. This time I use the sampsize argument which specify the numbers and the classes of the individuals taken for each tree. Thus our random forest is less biased by the -50K individuals.

```
test_clean_df2 <- test_clean_df[featuresToKeep]

test_clean_df2$income_predicted <- predict(rf8, test_clean_df2)

xtab <- table(test_clean_df2$income_predicted, test_clean_df$income)
conf_object <- confusionMatrix(xtab)

ber_final <- ber(t(conf_object$table ))

ber_final
```

```
## [1] 12.77779
```

After some tries I managed to get 12,80% of BER on the test set (same on the train data since random forests usually don't overfit). I had to do a compromise between a good classification of the class +50K and a less performant classification of class -50K which is usually very good.

Working with unbalanced dataset is a major challenge. This was the major difficulty of this test. There was also a big audit of the variables in order to remove the useless columns and perform some features engineering. If I had more time I would have done more precise tests to get with precision the level of correlation between variables (Chi2 test for example). I would also have provided a cleaner and factorized code. The dataiku challenge asked for one or two models. Here I performed only one, but this problem motivated me to go deeper in the exploitation of unbalanced dataset. Therefore I wrote the first equations to implement a logistic regression based on the Balanced Error Rate. The first step is to derive a Gradient and a Hessian of the BER. The second step is to prove the convexity of this new cost function. You can see that in the paper I gave you by email. I will then updtate it as a Latex document and implement it on Python or R.