IN-STK5000 – Adaptive Methods for Data-Based Decision Making
Project: Credit risk for mortgages - part 2

Luca Attanasio, S M Mamun Ar Rashid

September 2019

1 Introduction

How desirable would it be to use this model in practice? (focus on issues of reproducibility, reliability, privacy and fairness)
As explained in the previous assignment, two models were developed to choose between granting a loan or not: a neural network classifier and a random forest classifier. The best performing model in our case is the random forest classifier; nevertheless, its performances (75% average accuracy on the test set) are still far from reaching the perfect behaviour Figure 1, Figure 2. The interest rate is set to 5% and the results strongly depend on this parameter as they should. Indeed, by increasing the interest rate to 10%, all loans are granted and the behaviour is closer to the perfect banker. The behaviour is also much much better than the random banker with respect to the interest rate set to 5%. If we lower the interest rate to 1%, however, our bankers' results are negative and our decision would entail refusing to grant any loan. Yet, the two models are still performing better than the random banker.

2 Part 1

A Is it possible to ensure that your policy maximises the revenue?

As explained in the previous assignment, in order to ensure that our policy maximises the revenue, we can take the maximum between the expected utility of granting the loan and of not granting it. The corresponding action is our result. The value of

$$\max\left(E(U|A), E(U|B)\right) \tag{1}$$

is 0 if $E(U|A) < 0$ and $E(U|A)$ otherwise.

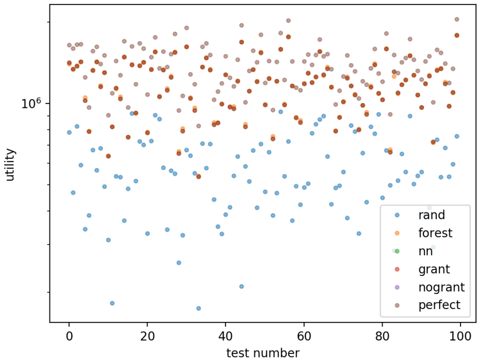

Figure 1: 100 individual tests. **Explanation:** In one test, the dataset is split randomly in two sets: the training set (80%) and the testing set (20%). The plot shows the utility on each test for each model. The utility values are different since the split is random in each test.

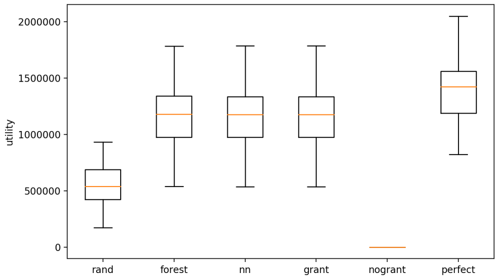

Figure 2: 100 tests boxplot. **Explanation:** The boxplot shows the variability of the utility function in the test set, when using random samples taken from the dataset for training (80%) and testing (20%). Using boxplots for the utility in a model, taken individually, we can estimate the mean and distribution of data in 100 tests.

In particular, the formula used to get the expected utility is the following, in case we grant the loan for a single sample $\vec{x}$:

$$E(U|A) = m \cdot (1 + r)^n \cdot (1 - P(\vec{x})) - m \cdot P(\vec{x}) \tag{2}$$

where m is the amount of the loan, $r = 5\%$ is the interest rate (per month), n is the lending period (in months) and A is the action of granting the loan.

If we do not grant the loan, then the expected utility is null:

$$E(U|B) = 0 \tag{3}$$

where B is the action of not granting the loan.

B How can you take into account the uncertainty due to the limited and/or biased data?

If the data is limited or biased, the bootstrapping technique can be taken into consideration. Bootstrapping allows to estimate the uncertainty or sensitivity of the algorithm related to the data. The testing accuracy of the classifier is not the actual expected performance because it is an unbiased estimate. A better estimate is available via bootstrapping. By taking each sample from the testing set and calculating the accuracy individually, we obtain an empirical distribution of scores. By averaging the scores, we obtain a better estimate of the test score. In addition, we may use multiple samples from the training data to get a more stable model.

The following plots are obtained on a single test (fitting the model with a single dataset split) by using the bootstrapping technique to get multiple prediction scores Figure 3 Figure 5. 1000 bootstrap samples were used in the evaluation. This testing allowed us to understand that the random forest is probably overfitting (although the decision trees are less prone to overfitting with respect to other models). If we train the model with different parameters: $max\_depth = 10$, $max\_features = 20$ then the model scores are more spread around the mean and the model performs better in the task of maximizing the utility Figure 4.

C What if you have to decide for credit for thousands of individuals and your model is wrong?

Our model may happen to be wrong. Depending on what goes wrong, the model can be changed to ensure it gets better results. Nevertheless, this can only be done *a posteriori*. Some precautions can be taken to avoid everything from going wrong. For example, we could train our model using cross validation to find the best parameter set for the model. Another method could be to merge other datasets from different banks to ensure that the training set is large enough so as to consider a greater number of cases (e.g. in one country salaries may be higher than in others and this affects the granting decision-making process). In addition, class weight and other techniques such as feature selection and feature scaling may help to get a more reliable model.

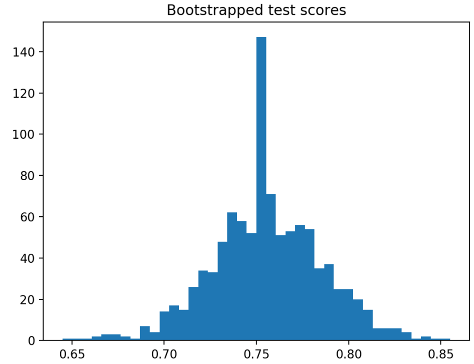

Figure 3: Bootstrap histogram on testing set for the random forest. $max\_depth = 15$, $max\_features = 35$

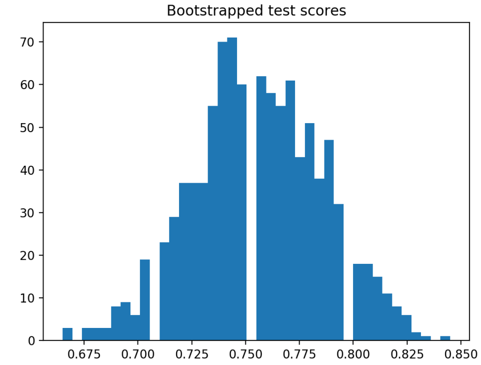

Figure 4: Bootstrap histogram on testing set for the random forest. $max\_depth = 10$, $max\_features = 20$

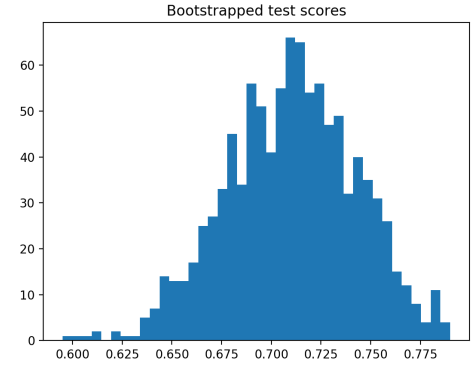

Figure 5: Bootstrap histogram on testing set for the neural network.

D How should you take that type of risk into account?

The empirical risk is evaluated when the accuracy of the classifier is tested. In our case, the classifier has 75% of average accuracy on the testing set. This does not mean that it chooses the wrong action 3 times out of 4, because we defined a utility function that ensures we can maximize the revenue. As a matter of fact, that type of risk can be taken into account in the policy. We need to make sure that we are not granting the loan if the probability that the user is not paying back is high.

For instance, if we notice that a great deal of people cannot afford paying back the loan, this may be the result of an ongoing economical crisis. As a consequence, it would be better to avoid granting most of the loans.

3 Part 2

Q: *Does the existence of this database raise any privacy concerns?*
The answer to this question is positive, if we wish to publish a database, we need to protect the identities of the people involved.

The easiest way to achieve privacy is to delete any information from it via *anonymisation*. In the credit risk database example, we need to erase the identity of the person. However, this may not be enough, especially in case of databases where the users identity is revealed and this can be joined with the credit risk database so that the attacker might link each field of the credit risk database to the identity of the person. This is achievable via *quasi-identifiers*: common properties from the samples in two or more databases that allow record linkage. In the credit risk example, the identities of the people had already been removed.

Furthermore, we need to ensure that some fields are replaced into ranges, to guarantee *k-anonymity*.

Q: *If the database was secret (and only known to the bank) but the credit decisions were public, how would that affect privacy?*
In this case, it would be impossible to reconstruct the data. In particular, if no column is leaked from the dataset and especially in case the data were shuffled, consequently it would be impossible to reconstruct the data. If we reverse the process of choosing the best action, we only know that the action assigned by the model was the best if it maximizes the expected utility $U(r)$. In other words, the loan was granted if:

$$E(U|A) \geq E(U|B) \tag{4}$$

The loan was not granted if:

$$E(U|A) < E(U|B) \tag{5}$$

where A is the action of granting the loan, B the action of not granting the loan. The utility function can be calculated in a variety of ways, but if the well-known function for calculating the amount that can be gained is used:

$$E(U|A) = m \cdot (1+r)^n \cdot (1 - P(\vec{x})) - m \cdot P(\vec{x}) \tag{6}$$

This function depends on 4 unknown parameters (r, m, n, $P(\vec{x})$), and since the value of $E(U|A)$ is positive when granting the loan, negative when not granting the loan, then the information is insufficient to solve the system. In addition, $P(\vec{x})$ is strictly dependent on the classification model, r, n or m could be discovered individually if all the other parameters are known. It can be noticed that $\vec{x} = (m, n, ...)$.

In our problem, we also know that if we do not grant the loan:

$$E(U|B) = 0 \tag{7}$$

These are the only information items we could gather, if we reverse engineer the problem and know how the utility function was created.

A How would you protect the data of the people in the training set?

In this case we do not know which could be *quasi-identifiers*, but some features can be selected and processed for research purposes. To protect the data of people in the training set, the following algorithm π can be used:

- Age: The age of a person can be quantified so that it is set to its closest set of ten. For example, 62 becomes 60, 69 becomes 70.
- Duration: The same strategy can be applied to the duration value. The value of the duration is set to the closest number divisible by 2. For example, 13 becomes 12, 12 stays 12.
- Amount: three strategies are implemented. The first involves the use of the *Local privacy model*, the second involves the use of the *Centralised privacy model*, the third uses a gamma distribution to fit the histogram of the amount function.

- Binary features: To guarantee better privacy, 30% of the dataset rows are chosen randomly and a default value of 1 is set for these rows in the corresponding binary feature column. This procedure is independent for each feature.

- Features from 1 to 4 (untouched): To ensure privacy, the nearest value to the one of the dataset rows chosen randomly can be set. For example, the value of 3 can be set to either 2 or 4 or be unchanged.

- Feature removal (unused): if our task is to predict whether we need to grant the loan or not, we could remove many features from the dataset based on a tree-based feature selection. This technique computes features' importance and discards irrelevant features.

The three strategies for the amount value are explained below:

1. One privacy model to interfere on the amount value is the *Localised privacy model.* In this model, the value of epsilon is set to $epsilon = 0.1$. By changing the value of epsilon for the amount value and keeping the same mechanism explained above for all other attributes, the utility will not have significant variations. In Figure 6, the definition of differential privacy is tested on each sample, individually. In our case, $\vec{x}$ is an attribute column of a sample taken from the dataset, $\vec{x}'$ is its modified version and ϵ is the value set to 0.1.

   $$max(ln\frac{\pi(a|\vec{x})}{\pi(a|\vec{x}')}) < \epsilon \tag{8}$$

   The model ensures there is at least a differential privacy with $\epsilon = 8$ with respect to the amount value, which is the maximum value in the boxplot Figure 6. The least differential privacy of ϵ for each technique is evaluated by taking the maximum of the fraction:

   $$\epsilon = max(ln\frac{\pi(a|\vec{x})}{\pi(a|\vec{x}')}) \tag{9}$$

   . The mean (orange horizontal line) is the value of a, defined during the lessons.

2. Another model that can be used is the *Localised privacy model.* In Figure 7, the utility is evaluated with respect to an increasing value of epsilon. The manipulation of other attributes is explained above. In the plot, by increasing the value of epsilon, more privacy is granted and the utility decreases.

3. An interesting method to ensure differential privacy is to fit the histogram of the amount value with a distribution (e.g. gamma) and to set the value on the dataset to the closest one on the fitted distribution Figure 8. This ensures privacy for the amount function. The fitting (amount error) ensures a differential privacy with at least $\epsilon = 18$ as shown in Figure 6. The error with respect to the dataset is large and this method has not been analyzed anymore.

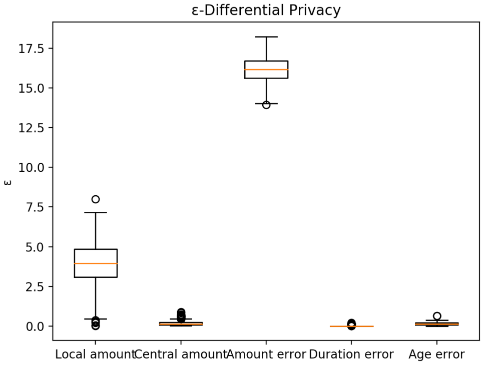

Figure 6: Differential privacy distribution for the different privacy techniques on each sample of the dataset. $\epsilon = 0.1$ in both Local amount and Central amount.

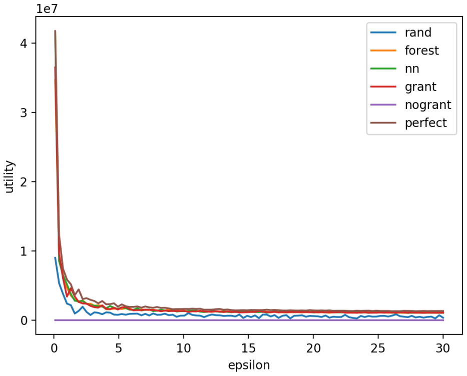

Figure 7: Utility with respect to epsilon (*Localised privacy model*).

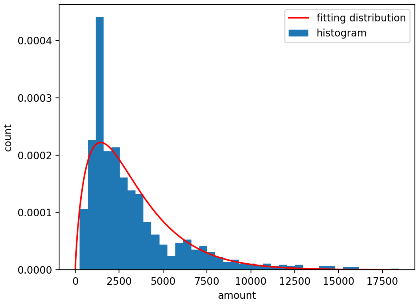

Figure 8: Fitting gamma distribution to amount.

B How would you protect the data of the people that apply for new loans?

In order to protect the data of the people that apply for new loans, it would be worth collecting the true data from the people and then process them to allow privacy guarantees. This ensures that there is always a most reliable dataset, useful for the company who gathered the data. If the dataset became public, then the data would be processed, as explained in the previous paragraph.

C Implement a private decision making mechanism for subsection B and estimate the amount of loss in utility as you change the privacy guarantee.

After adding privacy as explained above, the training and testing on the algorithm results for the utility do not vary significantly for the task of predicting good or bad loans in case we use the *Localized privacy model* with any ϵ. If we use the *Centralised privacy model*, the privacy strongly depends on ϵ and a higher value of it means more privacy but less utility. The best option is to select a *Localised privacy model*, using a low value for ϵ that guarantees privacy and a similar utility to the default one.