DISCUSSION PAPER SERIES

DP19105

TESTING FOR ASYMMETRIC INFORMATION IN INSURANCE WITH DEEP LEARNING

Serguei Maliar and Bernard Salanié

INDUSTRIAL ORGANIZATION

CEPR

ISSN 0265-8003

TESTING FOR ASYMMETRIC INFORMATION IN INSURANCE WITH DEEP LEARNING

Serguei Maliar and Bernard Salanié

Discussion Paper DP19105
Published 23 May 2024
Submitted 29 April 2024

Centre for Economic Policy Research
33 Great Sutton Street, London EC1V 0DX, UK
Tel: +44 (0)20 7183 8801
www.cepr.org

TESTING FOR ASYMMETRIC INFORMATION IN INSURANCE WITH DEEP LEARNING

Abstract

The positive correlation test for asymmetric information developed by Chiappori and Salanié (2000) has been applied in many insurance markets. Most of the literature focuses on the special case of constant correlation; it also relies on restrictive parametric specifications for the choice of coverage and the occurrence of claims. We relax these restrictions by estimating conditional covariances and correlations using deep learning methods. We test the positive correlation property by using the intersection test of Chernozhukov, Lee, and Rosen (2013) and the 'sorted groups' test of Chernozhukov, Demirer, Duflo, and Fernandez-Val (2023). Our results confirm earlier findings that the correlation between risk and coverage is small. Random forests and gradient boosting trees produce similar results to neural networks.

JEL Classification: C2, D8

Keywords: Insurance, Asymmetric information, Machine learning

Serguei Maliar - maliars@stanford.edu
Santa Clara University

Bernard Salanié - bs2237@columbia.edu
Department of Economics, Columbia University and CEPR

Acknowledgements
We thank Simon Lee and Vira Semenova for their very helpful comments and suggestions. We are grateful to Marc Maliar for his help with writing the TensorFlow code for estimation and testing, and to Xiangru Li for excellent research assistance.

Testing for Asymmetric Information in Insurance with Deep Learning *

Serguei Maliar[†] **Bernard Salanié**[‡]

April 26, 2024

Abstract

The positive correlation test for asymmetric information developed by Chiappori and Salanié (2000) has been applied in many insurance markets. Most of the literature focuses on the special case of constant correlation; it also relies on restrictive parametric specifications for the choice of coverage and the occurrence of claims. We relax these restrictions by estimating conditional covariances and correlations using deep learning methods. We test the positive correlation property by using the intersection test of Chernozhukov, Lee, and Rosen (2013) and the "sorted groups" test of Chernozhukov, Demirer, Duflo, and Fernández-Val (2023). Our results confirm earlier findings that the correlation between risk and coverage is small. Random forests and gradient boosting trees produce similar results to neural networks.

*We thank Simon Lee and Vira Semenova for their very helpful comments and suggestions. We are grateful to Marc Maliar for his help with writing the TensorFlow code for estimation and testing, and to Xiangru Li for excellent research assistance.

[†]Santa Clara University.

[‡]Department of Economics, Columbia University, bsalanie@columbia.edu.

1 Introduction

In insurance markets, the *positive correlation property* (PCP) states that insurees' choices of coverage should be positively related with ex-post measures of their risk, such as the occurrence and severity of claims. This can be either because of adverse selection (i.e., riskier insurees self-select into contracts with higher coverage, as in Rothschild and Stiglitz (1976)), or because of moral hazard (i.e., a higher coverage discourages prevention effort). Chiappori, Jullien, Salanié, and Salanié (2006) proved that the PCP obtains quite generally in models of competitive equilibrium in insurance markets.

Chiappori and Salanié (2000) used the PCP to propose a test for asymmetric information in insurance; and they applied it to French car insurance data. In its simplest form, the PCP states that the conditional correlation of coverage and ex-post risk should be positive, for all values of the vector of covariates that are observed both by the insurer and the insuree[1]. Suppose that coverage is treated as a binary choice (i.e. minimal versus comprehensive); and that ex-post risk is also binary (e.g. whether or not the insuree filed a claim). Chiappori and Salanié (2000) argued that under the PCP, the residuals of two binary choice models for coverage and ex-post risk should be positively correlated, if all public covariates are controlled for. Hence, they tested the hypothesis that the correlation of the generalized residuals of two univariate probits was zero; they also estimated and tested for the correlation of the generalized residuals of a bivariate probit.

To their surprise, Chiappori and Salanié (2000) found no statistical evidence for the PCP with any of these tests in the French car insurance data they were using: the correlation of coverage and risk was close to zero.[2] This remarkable fact implies that essentially all relevant information is contained in publicly observed covariates.

However, the analysis of Chiappori and Salanié (2000) relied on restrictive specifications that may have limited power to detect the PCP. They only included about fifty regressors in their probit regressions, which was a very small subset of the covariates they could have constructed by interacting the covariates. Moreover, their bivariate probit model was built on a simplifying assumption of constant correlation. [3].

[1] As explained in Chiappori, Jullien, Salanié, and Salanié (2006) and Chiappori and Salanié (2013), the precise statement of the positive correlation property needs to be adapted in more general models.

[2] This is not a universal finding; while the PCP was not documented with car insurance data, there is evidence for the PCP in some other markets. Our purpose is not to discuss the PCP evidence, *but to explore how the outcome of the PCP test depends on design and implementation of the testing procedure.*

[3] The test procedures in Chiappori and Salanié (2000) have been extended in several directions. Kim, Kim, Im, and Hardin (2009) showed how the probit for coverage can be replaced by an ordered multinomial choice model when more than two types of contracts are available. Chiappori and Salanié (2000) had also proposed a fairly basic nonparametric test. Following ideas in Dionne, Gouriéroux, and Vanasse (2001, 2006), Su and Spindler (2013) and Spindler (2014) used a more powerful nonparametric test of conditional independence.

The extraordinary development in machine learning methods in the past decade suggests revisiting some seemingly well-established empirical findings. Our goal here is to show how these methods can be implemented in the context of the PCP, and to check whether more powerful testing methods can alter the conclusions of Chiappori and Salanié (2000) about its quantitative unimportance.

Ideally, one would want to do two things: estimate flexibly the conditional correlation of risk and coverage for any given values of the covariates, and test that this conditional correlation is positive for all values in a given subset (e.g. for all male, 40- to 45-year old drivers who use a 5-year old car). We show that both goal can be achieved by combining the flexibility of machine learning methods and standard econometric tests. From a growing catalog of machine learning methods on the market, we choose deep learning as our main estimation method because of its popularity and its remarkable success in many applications.

For the sake of comparison, we use the same car insurance data as in Chiappori and Salanié (2000). The dataset contains only 6,333 observations, a typical size for many microeconometric applications. This allows us to explore the effectiveness of deep learning techniques in such settings. Each observation includes information on the car (brand, model, age, power, ...) and the client's demographics (age, profession, residence,...); we use these variables and their interactions to construct the covariates x. We summarize coverage c and ex-post risk r by two binary variables. Let X denote all publicly observed covariates. We denote $p_{jk}(x)$ the probability that $c = j$ and $r = k$ given $X = x$, for $j, k = 0, 1$. Finally, we let $p(x) = p_{10}(x) + p_{11}(x)$ (resp. $q(x) = p_{01}(x) + p_{11}(x)$) denote the probability that $c = 1$ (resp. that $r = 1$) conditional on $X = x$. The probability $p(x)$ is the conditional choice probability of the higher coverage, and $q(x)$ is the conditional probability of an at-fault claim.

The most basic form of the positive correlation property states that for all values of x, the covariance of c and r conditional on $X = x$ is non-negative:

$$C(x) \equiv \operatorname{cov}(c, r|X = x) = p_{11}(x) - p(x)q(x) \geq 0.$$

A shortcoming of the covariance is that its value is not easily interpretable. We therefore also state the positive correlation property in terms of the correlation coefficient:

$$\rho(x) \equiv \frac{C(x)}{\sqrt{p(x)(1-p(x))q(x)(1-q(x))}} \geq 0.$$

Estimating the covariance and correlation functions for given values of the covariates $X = x$

However, this is only practical when there are no more than three continuous covariates and a small number of discrete covariates.

requires estimating the p_{jk} probabilities flexibly. This is not a simple task, as many interactions between the covariates can have explanatory power. It is notoriously hard, for instance, to model the risk $q(x)$ parsimoniously. It is even more difficult to test that ρ is non-negative over a subset of covariates: interesting subsets typically are very large, leading to a multiple testing problem where the distributions of the estimated covariances or correlations for different x are not independent.

We implement three approaches to apply machine learning to testing for the PCP. Our first test relies on a feedforward neural network to predict the conditional probabilities $p_{jk}(x)$: in the terminology of this field, this is a 4-way classification problem. A potential complication we face when testing the positive correlation property is that the neural network estimates $\hat{p}_{jk}(x)$ have a relatively slow rate of convergence and act as nuisance parameters. As it turns out, the covariance function has a nice double robustness property; the presence of these nuisance parameters is not an issue. On the other hand, the presence of such nuisance parameters does complicate inferences about the correlation function.

To remedy this, we use the double-debiasing method of Chernozhukov et al. (2018), which extends the idea of Neyman orthogonalization to a broad range of models and estimation methods. We combine this double-debiasing method with results from Semenova and Chernozhukov (2021) to obtain consistent and asymptotically normal estimators of the average values of the covariance and correlation function within groups of observations. We then use the intersection tests developed in (Chernozhukov, Lee, and Rosen (2013)) to test the positive correlation property for a variety of groups of observations. We test, for instance, that the correlation is positive on average for all modalities of the "age of the car" variable.

We find that the neural network predicts the purchase of coverage considerably better than it does the accident occurrence. This is not that surprising: at-fault claims are relatively low-probability events and insurers know that they are hard to predict. The range of the estimated covariance function lies mostly in the interval $[-0.01, 0.01]$. While the correlation function has a larger range, it narrows considerably when we double-debias it and we average within groups. Our intersection tests show that for any of our eight covariates, we can reject the hypothesis that the correlations are positive on average for all values of its modalities. On the other hand, we cannot reject the hypothesis that these average correlations are larger (i.e., less negative) than a small negative number like -0.05. In the end, we obtain a 95% confidence interval for the range of these average correlations that is confined to a narrow interval around zero.

Our second method relies on the "sorted groups" approach of Chernozhukov, Demirer, Duflo, and Fernández-Val (2023). We first cross-fit the neural network model that we selected in our first approach. We then allocate observations into groups sorted by the value of the predicted covariance or correlation. To maximize the power of the test of the PCP, we focus on the

observations for which the estimated correlation is smallest in algebraic terms. We find here again that only very small values of the covariance and correlation are consistent with the data, and we find no significant evidence for the PCP.

Finally, we run two variants of our first approach in which we replace the neural network with two other popular machine learning methods—random forests and gradient-boosted trees. While these two methods put different weights on the various covariates, the results of the intersection tests are similar to those obtained by using our baseline deep learning method.

To summarize: machine learning methods do not alter the qualitative conclusion of Chiappori and Salanié (2000) that the correlation of coverage and risk is essentially zero. Even more remarkably, we could not find evidence for a positive correlation in any reasonably-sized subpopulation. While adverse selection and moral hazard are clearly important phenomena in many markets, they do not seem to play much of a role in this one.

The rest of the paper is as follows. Section 2 describes the data used for estimation. Our analysis has three steps: In Section 3, we fit a neural network to classify insurees into the four alternatives $c, r \in \{0, 1\} \times \{0, 1\}$. In Section 4, we use the double-debiasing methods of Chernozhukov et al. (2018) to correct the correlation function. In Section 5 we test the positive correlation property by running the intersection test of Chernozhukov, Lee, and Rosen (2013). Finally, Section 6 compares our results with those obtained using methods based on decision trees.

2 The Data

Chiappori and Salanié (2000) obtained their data from the French federation of insurance companies FFSA, which ran a survey of automobile insurance in 1989. They selected a subsample that only includes "young" drivers—insurees who obtained their driver's licence within the past three years. We focus on an even narrower subsample of insurees whose driving license is one year old at most. Since these individuals have no previous driving history, there is no concern about the impact of experience rating on driving behavior; it also reduces the unobserved heterogeneity in the sample.

Our selection leaves us with a sample of 6,333 observations. The data on each insuree and car are quite rich. Each observation includes information on the car (brand, model, age, power, ...), the client's demographics (age, profession, residence,...), the type of contract, and the claim record. As in Chiappori and Salanié (2000), we code the type of contract and the claim record as binary.

The data also record if the insurance contracts covered all or part of the year. Only about 40% of insurees were insured throughout the year, and roughly 15% were covered for less than

two months. Like in Chiappori and Salanié (2000), we use sampling weights w that represent the number of days that the insuree was insured during the year. We indicate sampling weights with w subscripts when needed.

In automobile insurance, the main distinction between contracts is whether they only include third-party (liability) coverage—which is compulsory in France—or whether they also cover damages that the insuree caused to his/her own car. We call the latter "comprehensive coverage", and we will neglect variations within this class, such as the amount of the deductible. Since the difference between third-party and comprehensive coverage only matters when the insuree is at fault, we define our claim variable accordingly. This results in the two binary variables $c, r \in \{0, 1\}$:

- $c = 1$ if the insuree opted for comprehensive coverage
- $r = 1$ if the insuree filed at least one at-fault claim.

We also define four indicator variables as $y_{jk} = 1$ if ($c = j$ and $r = k$) for $j, k = 0, 1$. Table 1 classifies the 6,333 observations by the four indicators.

Event	c	r	Alternatives	Number of observations
Third-party, no accident	0	0	$y_{00} = 1$	3,696
Third-party, accident	0	1	$y_{01} = 1$	302
Comprehensive, no accident	1	0	$y_{10} = 1$	2,203
Comprehensive, accident	1	1	$y_{11} = 1$	132
Total				6,333

Table 1: Observed Classification

We use the same set of covariates X as Chiappori and Salanié (2000); they are created from the eight variables that insurers identified as being the most important. We have up to 28,800 categories of insurees: nine age categories, eight professions, four types of use, ten regions, five rural-to-urban codes, and gender; and 72 car categories: six categories for the performance of the car, and twelve for its age. Combining them would yield more than 2 million dummy variables, a number that dwarfs the sample size. Even if we had a much larger dataset, there are many more variables in the data. This is a clear-cut "$p \gg n$" case, which calls for model selection.

3 The Deep Learning Model

We fit the 4-way probabilities $p_{jk}(x) = \Pr(c = j, r = k|X = x)$ for $j, k = 0, 1$ with a neural network. The values of the eight covariates x enter the input layer. The neural network has D

hidden layers, each with W neurons; the last hidden layer feeds into a 4-node output layer which uses a "softmax" function to generate the probabilities $p_{jk}(x)$. We train the neural network using the Adam optimizer (Kingma and Ba, 2017) to minimize the cross-entropy loss function adjusted by the sampling weights. For a sample of observations $(x_i, c_i, r_i, w_i)_{i \in I}$, the loss is

$$L = \sum_{i \in I} w_i \sum_{j,k=0,1} \mathbb{1}(c_i = j, r_i = k) \log p_{jk}(x_i).$$

Our code relies on the Python package Keras (Chollet, 2021) with the TensorFlow backend (Abadi et al., 2016).

The class of neural networks we consider has $P \equiv n_X W + (D-1)W^2 + 3W$ parameters for $D > 0$, where n_X is the number of covariates in the input layer[4]. Adding up the number of categories (minus one category per variable) of our eight covariates, plus the constant, gives us $n_X = 49$ and results in $P = W(2 + (D-1)W)$ parameters. To illustrate, a very modest neural network with $D = 2$ hidden layers of $W = 16$ neurons each has $P = 1,088$ parameters.

With such a large number of parameters, overfitting is an obvious concern. To guard against it, we resort to cross-validation: we use a validation sample to decide when to stop learning. In addition, we randomly drop out some of the neurons in the hidden layers during training. The idea of dropout regularization, pioneered by Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014), is that the network will learn to compensate for the missing neurons, and that the resulting model will be more robust to overfitting. The fraction d of neurons that are dropped out is called the dropout rate. Dropout allows us to use a more powerful network with a larger number of neurons. The optimal dropout rate typically increases with the number of neurons in each hidden layer, as a higher dropout rate is needed to deal with higher overfitting.

3.1 Hyperoptimizing the Neural Network

The neural network we consider has many hyperparameters: its depth D and width W, the size of the validation sample, the size of the mini-batches, the choice of optimizer, etc. We decided to optimize over the depth and width, and also over the dropout rate d.

Since our sample size is relatively small, we only fit smallish neural networks that vary in:

- the number D of hidden layers: we tried 0, 1, 2 and 3 hidden layers;
- the number W of neurons in each layer: tried 8, 16, and 24 neurons in each layer;
- the dropout rate d: we tried values from 0 (no dropout) to 0.8.

[4]For $D = 0$, the number of parameters is $3n_X$.

To optimize in hyperparameter space, we split the data into training, validation and testing sets, representing 70%, 15%, and 15% of the data, respectively. For each 3-uple of values of D, W and d, we fit the neural network on the training sample and we use the validation sample to stop training soon after the loss on the validation sample stops decreasing. To discard local minima, we keep training the network for a small number of epochs to see if the validation loss keeps increasing[5]. We then store the parameters of the neural network that correspond to the epoch with the smallest validation loss. Then we measure the loss on the test sample.

After fitting all of these models, we select the combination of the three hyperparameters that leads to the smallest loss on the test sample[6]. The hyperoptimized model has $D = 2$ hidden layers; $W = 16$ neurons in each layer; and a dropout rate $d = 0.1$. The hyperoptimizing procedure took 150 seconds on a Mac Studio. The resulting neural network has 1,524 parameters.

We also fit a 2-way classification model for c and another for r; we hyperoptimized them in a similar way. The neural network for c (the choice of coverage) has no hidden layer and dropout = 0.7. The model for r (the occurrence of a claim) has 2 hidden layers; 8 neurons in each layer; and dropout = 0.4.

Variables	Constant	Probit	Neural network
(c, r)	0.963	0.629	0.441
c	0.673	0.385	0.256
r	0.283	0.244	0.180

Table 2: Comparing Losses

These loss values can be compared with those of probit models that use the same set of regressors as in Chiappori and Salanié (2000), as well as to "constant" models that use no regressors. As Table 2 shows, our deep learners massively outperform the probits of Chiappori and Salanié (2000). Since the loss is just minus the average log-likelihood per observation, these improvements are quite large. For instance, the 0.188 gain in average log-likelihood from bivariate probit to the bivariate deep learner yields a likelihood ratio statistic of $2 \times 6,333 \times 0.188 = 2381.2$, for an additional $1,524 - (2 \times 48 + 1) = 1,425$ parameters. The corresponding p-value is minuscule[7].

[5]The parameter that controls the number of periods that we wait is called "patience"; we set it at 10 epochs.

[6]Fitting a neural network involves a choice of initial values for the weights. TensorFlow uses a well-tested procedure that injects some randomness into the process. As a result, different runs can select slightly different neural networks. We find that this had almost no impact on the results of our final tests of the positive correlation property.

[7]This is only meant to be illustrative; it is not clear that one can use asymptotic approximations with such a large number of parameters.

We also applied our procedure to the sampling weights. We use an activation function that takes into account the truncation of weights in $[0, 1]$ and a loss function that allows for a mass point at $w = 1$. Not surprisingly, the selected neural network is rather sparse: it has no hidden layers ($D = 0$) and a dropout rate $d = 0.5$. The whole hyperoptimizing procedure took 130 seconds. The hyperoptimized model has a loss of 0.098 on the test sample, while a constant model has a test loss of 0.114, which is only 15% larger than the model.

3.2 Estimated probabilities and weights

Figure 1 plots the estimated probabilities $\hat{p}_{jk}$, both when the corresponding y_{jk} is 0 and when it is 1. The dashed red line represents the mean of y_{jk} in the sample. A perfect fit would have $\hat{p}_{jk} = y_{jk}$ within each panel. It is clear that the left column performs better than the right column in this respect. It is not that surprising as there are more than 15 times as many observations with $r = 0$ as with $r = 1$: the neural network puts a large weight on fitting these observations. Figure 2 shows that the model can predict the purchase of coverage considerably better than the accident occurrence. This is a common finding with insurance data. It is more surprising that the neural network overestimates the probability of a claim to the extent shown in the right panel. This seems to be a side effect of its efforts to fit the choice of contract (the variable c). For comparison, Figure 2 shows the results obtained with the 2-way classification neural networks for c and for r.

Figure 3 plots the fitted weights. As already mentioned, the model for weights has low explanatory power. Many contracts cover the entire year, as can be seen from the vertical cluster on the right. The covariates do not help much in predicting how long the car is insured within a given year, as it depends on decisions to buy, sell or exchange a vehicle that result in starting or terminating coverage within a given calendar year.

In a linear model, we could use partial R^2's to evaluate the contribution of various covariates to explaining the left-hand side variable. In a neural network, a natural alternative is to train the model again while omitting one covariate and to measure the additional loss. This is a very partial indication, as the neural network interacts different groups of variables in potentially complex ways. Still, it is a reasonable starting point.

Our application has eight groups of variables; accordingly, we run eight neural networks, each of which omits one of these groups. Figure 4 plots the results. The dashed vertical line represents the test loss of the complete neural network, which is denoted "None". A larger positive number denotes that this group of variables contributes more to the quality of the fit (that is, reduces the test loss more).

Figure 4 shows clearly that the age of the car dominates the fit. Five other groups of

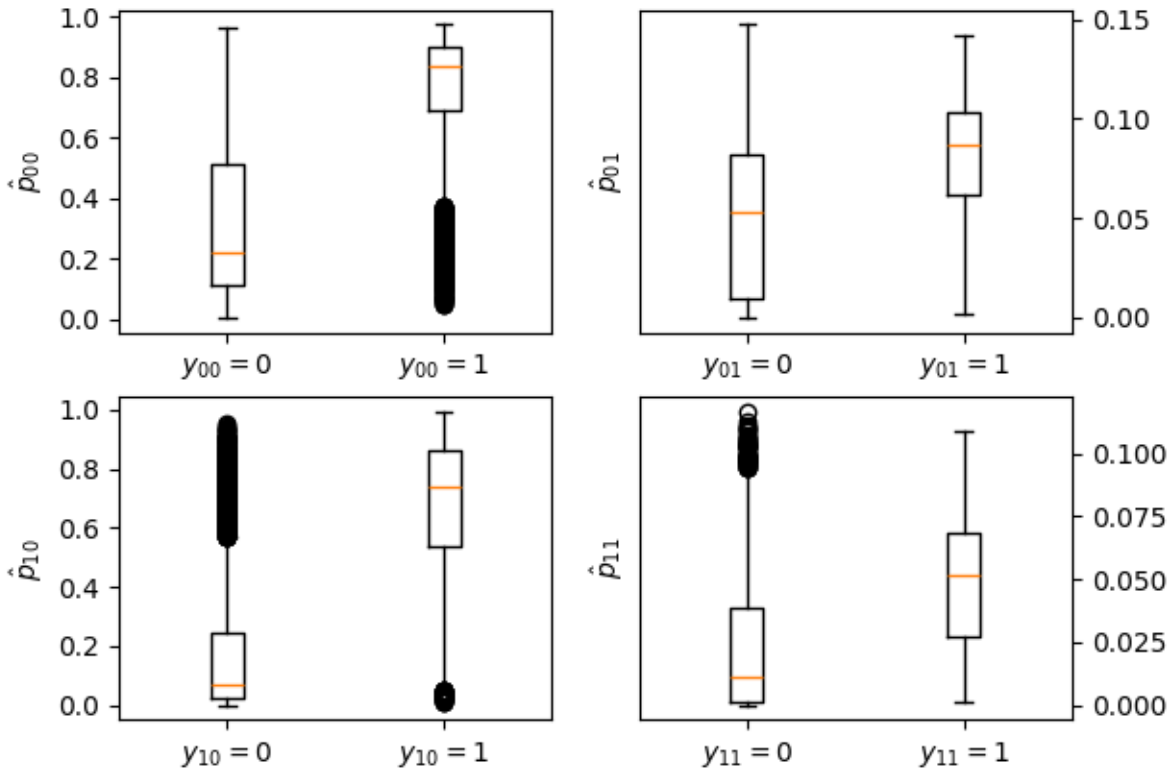


Figure 1: Fitting the y_{jk} variables

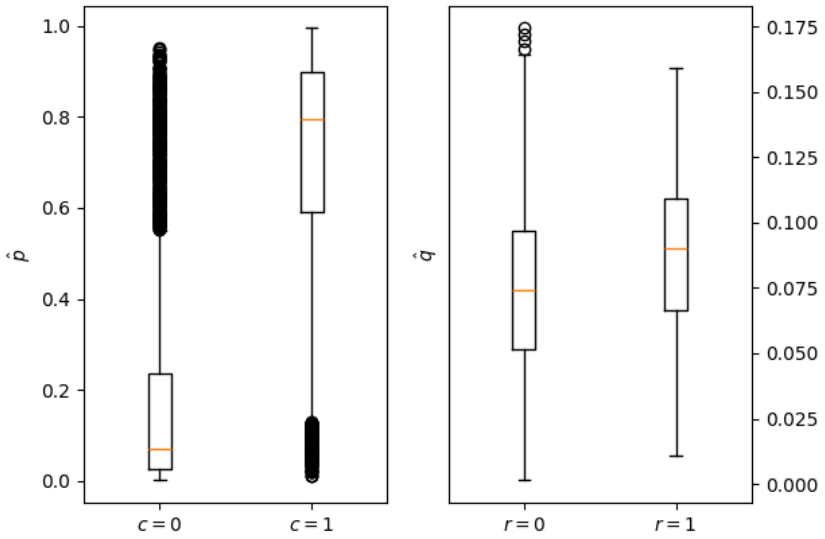


Figure 2: Fitting the c and r variables

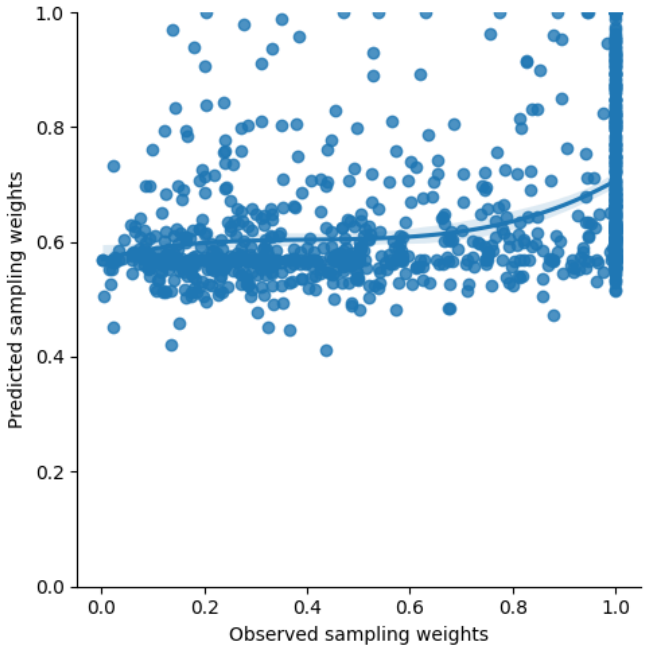


Figure 3: Fitting the weights w

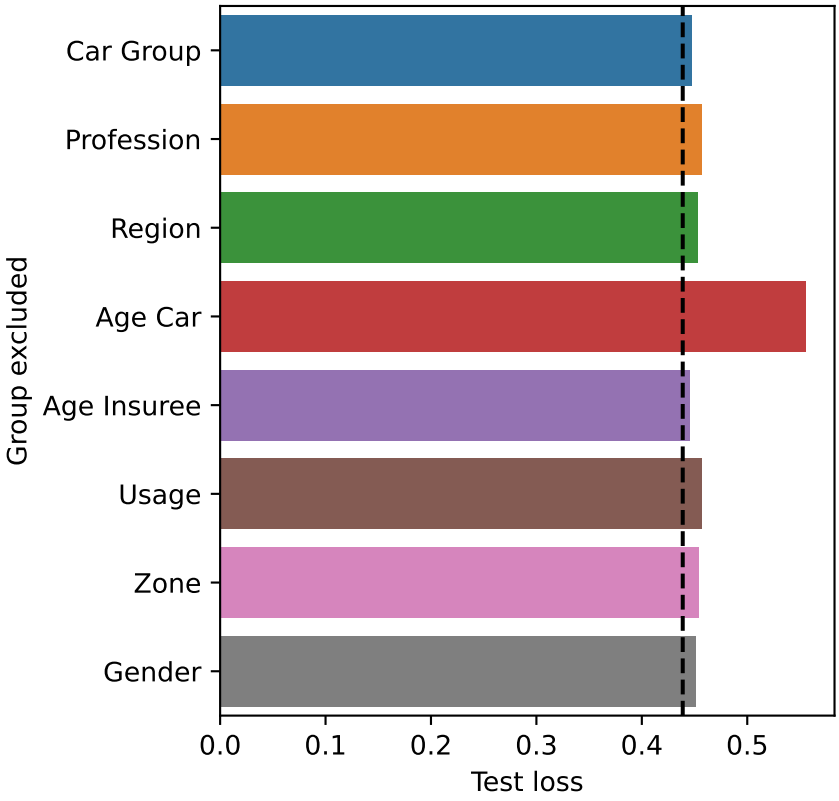


Figure 4: Omitting groups of variables

variables contribute (taken by themselves) to the fit. In decreasing order of importance, they are: the (work/leisure) usage of the car, the profession, the rural/urban zone, the age of the insuree, and her gender. Looking more closely at the models for c and r shows that drivers of older cars are less likely to buy comprehensive coverage—again, a common finding.

3.3 Estimated covariance and correlation

We use the hyperoptimized models of Section 3.1 to get "raw' and "cross-fitted" estimates for the covariances and correlation functions. The raw estimates are simply the values predicted over the whole sample. To obtain the cross-fitted estimates, we split the sample randomly into five subsets; we predict the covariances and correlations over a subset using the hyperoptimized neural network trained over the other four subsets only. We will need the cross-fitted estimates when we move to testing in Section 5.1.

Table 3 gives the results. Only 40.4% of the predicted raw covariances (and correlations) are positive. The cross-fitted estimates are very similar to the raw estimates, if somewhat more dispersed.

Name of variable	Raw estimates			Cross-fitted estimates		
	Mean	Dispersion	Range	Mean	Dispersion	Range
Covariance	-0.000	0.004	[-0.014, 0.014]	-0.001	0.003	[-0.019 , 0.016]
Correlation	-0.007	0.043	[-0.098, 0.092]	-0.013	0.033	[-0.121, 0.106]

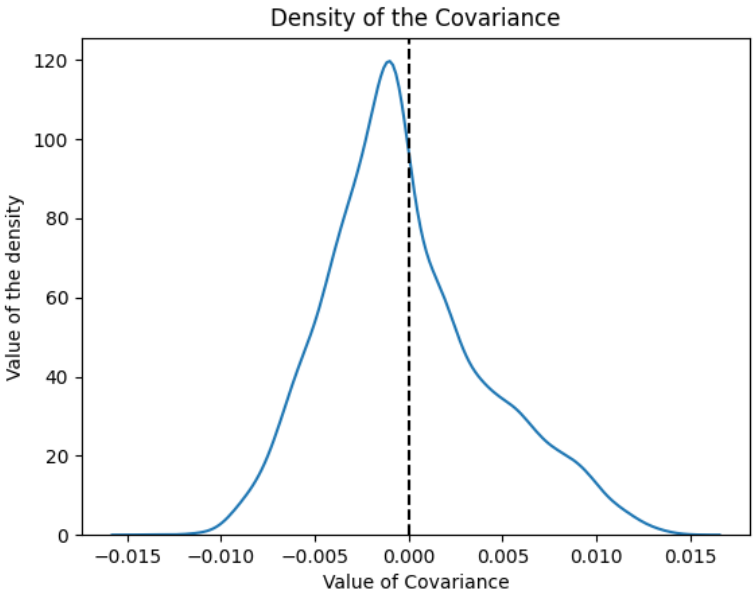
Table 3: Raw and cross-fitted neural network estimates for the covariance and correlation

Figure 5 plots the density of our estimated covariance function $\hat{C}(x)$ and correlation function $\hat{\rho}\,(x)$ over the sample.

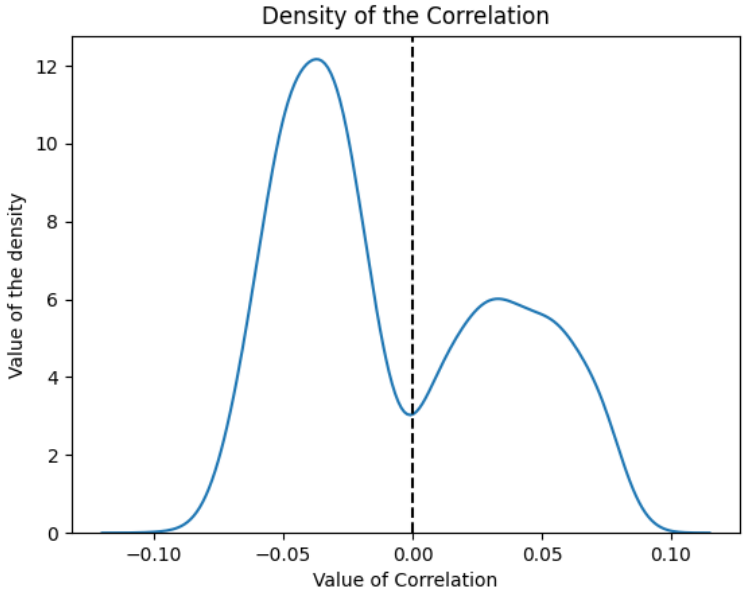
Essentially all the mass of the distribution of the covariance is situated in the interval $[-0.01, 0.01]$. These seem like small values, but they are not easily interpretable. We next consider the correlation function. It is negatively skewed; more than 60% of the estimates are negative. The correlations are small, however: 99% of the mass is in the $[-0.2, 0.2]$ interval.

It would be tempting to interpret the bimodal shape of the estimated density of $\hat{\rho}$ as a mixture of the densities for the two genders. We already know from Figure 4 that gender has low explanatory power, however. Figure 6 has boxplots of the distributions of $\hat{\rho}$ when a variable takes one particular value. The top-left panel, for instance, shows that the estimated correlation tends to decrease with the age of the car. Men seem to have a lower correlation of risk and coverage than women, rural drivers a lower correlation than urban drivers.

Figure 5: Densities of $\hat{C}(x)$ and $\hat{\rho}(x)$ over the sample (raw neural network estimates)



(a) Density of $\hat{C}(x)$



(b) Density of $\hat{\rho}(x)$

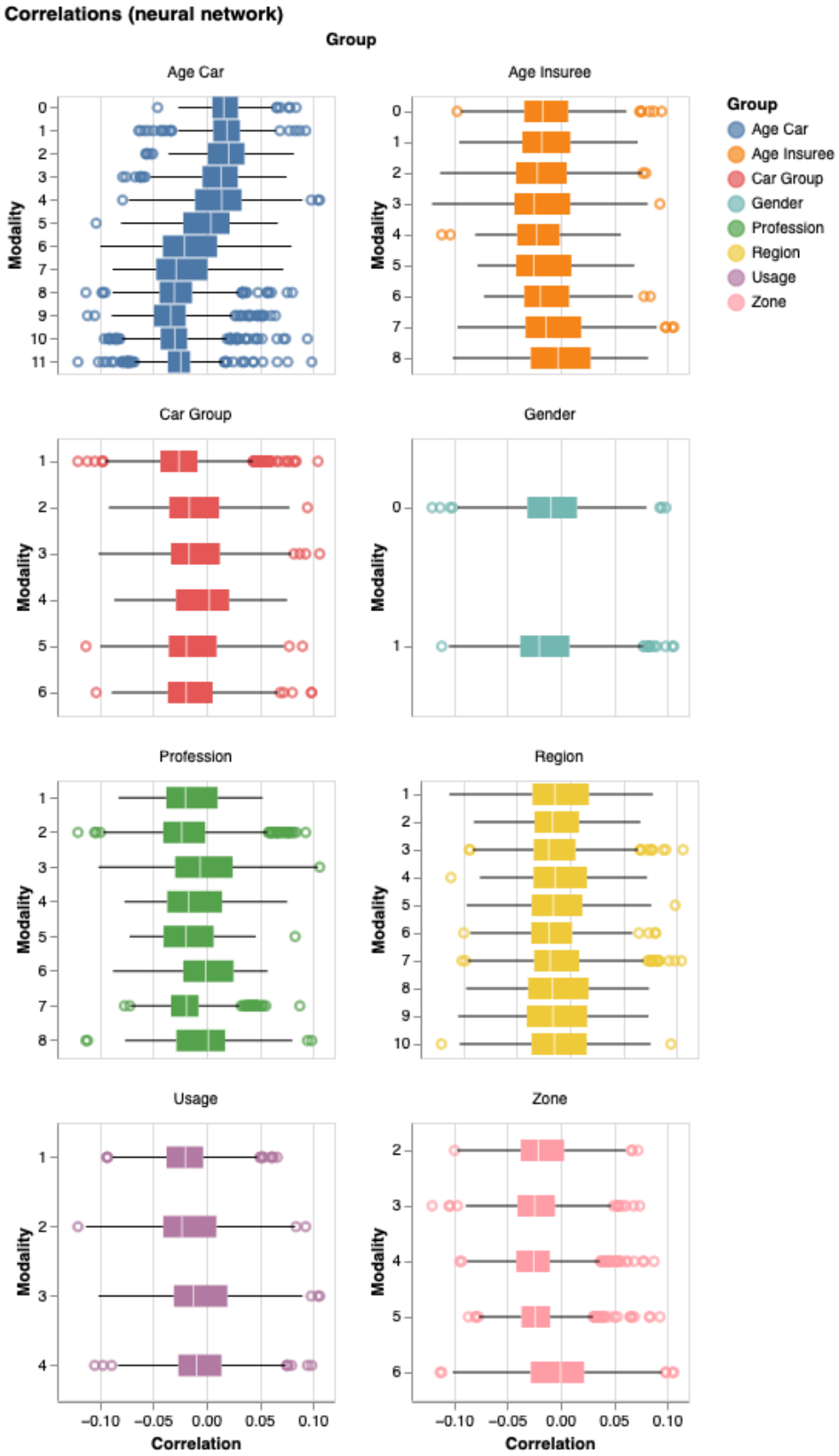


Figure 6: Correlation $\hat{\rho}(x)$ for different subgroups (raw neural network estimates)

4 Nuisance parameters and double-debiasing

Our ultimate goal is to test the sign of the covariance function $C(x)$ and the correlation function $\rho(x)$. These are easily estimated by plugging in the probabilities $\hat{p}_{jk}(x)$ predicted by the neural network. Still, these probability estimates have a relatively slow rate of convergence, which is likely to contaminate inference. This is a common issue with machine learning methods: they yield predictors that must be treated as nuisance parameters in later stages of statistical procedures. To remedy this problem, Chernozhukov et al. (2018) developed a double-debiasing method that extends the idea of Neyman orthogonalization to a broad range of models and estimation methods.

To give the intuition behind double-debiasing, consider the vector of probabilities $\eta \equiv (p_{00}, p_{01}, p_{10}, p_{11})$. We denote η_0 its true value. Suppose that we want to estimate some parameter vector β_0 that satisfies a set of conditions $M(\beta_0, \eta_0) = 0$. In our application, this β_0 will be the average covariance or correlation for a given subset of observations—for instance, for all young men.

The neural network (or any other estimation procedure) gives us an estimate $\hat{\eta}$. A natural way to proceed would be to estimate β_0 by the value $\hat{\beta}$ that minimizes some norm of $\hat{M}(\beta, \hat{\eta})$, where the function $\hat{M}$ is the sample analog of M. Given an appropriate set of assumptions, the standard Taylor expansion around the true values (β_0, η_0) gives

$$\nabla_\beta M(\beta_0, \eta_0)(\hat{\beta} - \beta_0) \simeq -M(\beta_0, \eta_0) - \nabla_\eta M(\beta_0, \eta_0)(\hat{\eta} - \eta_0). \tag{1}$$

The presence of the second term on the right-hand side is what makes the $\hat{\eta}$ estimates nuisance parameters: the estimation error $\hat{\eta}$ contaminates the asymptotic distribution of $\hat{\beta}$ if the gradient $\nabla_\eta M(\beta_0, \eta_0)$ is nonzero.

If the $\hat{\eta}$ estimates converge at the usual parametric rate, the additional term only changes the usual sandwich formula. This clearly does not apply here, as the neural network estimates converge more slowly, and this invalidates standard inference over β_0. To get rid of this nuisance effect, we need to change the estimating equation to an equation for which the gradient of M with respect to η is zero at the true values (β_0, η_0). This can be done by projecting the estimating equation on the orthogonal subspace to the gradient $\nabla_\eta \hat{M}$. This is the idea that underlies Neyman orthogonalization and its modern successor, double-debiasing.

We describe how double-debiasing applies to the covariance and the correlation functions in Sections 4.1 and 4.2, respectively.

4.1 Covariance

Given our estimates $\hat{p}_{jk}$ of the probabilities of the four alternatives, we can write the covariance as

$$\widehat{C}(x) = \frac{\hat{E}\left(w(c-\hat{p}(x))(r-\hat{q}(x))|X=x\right)}{\hat{E}(w|X=x)} \equiv \hat{E}_w\left((c-\hat{p}(x))(r-\hat{q}(x))|X=x\right),$$

where $\hat{E}$ (resp. $\hat{E}_w$) denotes an unweighted (resp. weighted) sample mean; $\hat{p}(x) \equiv \hat{p}_{10}(x)+\hat{p}_{11}(x)$ estimates $\Pr(c=1|X=x)$ and $\hat{q}(x) \equiv \hat{p}_{01}(x)+\hat{p}_{11}(x)$ estimates $\Pr(r=1|X=x)$. In these equations, the neural network estimates $\hat{p}_{jk}(x)$ act as nuisance parameters $\hat{\eta}$. However, it turned out that for the covariance coefficients, the presence of such parameters does not interfere in the estimation procedure, so that standard inference methods apply.

To see this, consider estimating the sample-weighted average covariance β_g for a group of observations $i \in g$. We will use weighted non-linear least squares. This corresponds to the estimating equation

$$M(\hat{\beta}_g, \hat{\eta}) = \sum_{i \in g} w_i \left(\widehat{C}(x_i) - \hat{\beta}_g\right) = 0.$$

The gradient of M with respect to η is a weighted sum of gradients of $C(x)$ with respect to η. Now consider, for some x and some $j,k = 0,1$,

$$\frac{\partial C(x)}{\partial p_{jk}(x)} = -\frac{\partial p(x)}{\partial p_{jk}(x)} E_w\left((r-q(x))|X=x\right) - \frac{\partial q}{\partial p_{jk}} E_w\left((c-p(x))|X=x\right).$$

By definition, at the true values η_0 we have

$$E_w\left((r-q(x))|X=x\right) = E_w\left((c-p(x))|X=x\right) = 0.$$

Since the partial derivatives of p and q with respect to p_{jk} are either 0 or 1, the gradient of M with respect to p_{jk} is zero at the true values and the estimates of $\hat{\eta}$ do not affect the asymptotic distribution of $\hat{\beta}$. To put it differently: the covariance is doubly robust and does not need to double-debiased.

4.2 Correlation

We now turn to the correlation function $\rho(x)$. A naive estimate of the correlation function would be

$$\hat{\rho}(x) = \frac{\widehat{C}(x)}{\sqrt{\hat{p}(x)(1-\hat{p}(x))}\sqrt{\hat{q}(x)(1-\hat{q}(x))}}.$$

The gradient of $\hat{\rho}$ with respect to $\hat{\eta}$ now involves terms like

$$\frac{\partial \sqrt{\hat{p}(1-\hat{p})}}{\partial \hat{p}_{jk}}$$

which are clearly not zero at the true values. Therefore the presence of nuisance parameters does interfere in the estimation procedure and thus, can invalidate inference unless we use double-debiasing.

Proceeding as with the covariance function, let our estimation equation be

$$M(\hat{\beta}_g, \hat{\eta}) = \sum_{i \in g} w_i \left(\hat{\rho}(x_i) - \hat{\beta}_g \right) = 0.$$

Again, the gradient of M with respect to η is a weighted sum of gradients of $\rho(x)$ with respect to $\eta(x)$. We already know that we the derivatives of $C(x)$ with respect to $\eta(x)$ are zero. This leaves us with

$$\frac{\partial \rho(x)}{\partial p_{jk}(x)} = \rho(x) \times \left(\frac{p(x) - 1/2}{p(x)(1-p(x))} \frac{\partial p(x)}{\partial p_{jk}(x)} + \frac{q(x) - 1/2}{q(x)(1-q(x))} \frac{\partial q(x)}{\partial p_{jk}(x)} \right).$$

To apply double-debiasing, we need to project our estimating equation on the orthogonal space to this gradient. This can be done quite simply by running a weighted regression of $\hat{\rho}(x_i)$ on the variables that correspond to $(j,k) = (0,1), (1,0)$, and $(1,1)$. In fact, the last one is the sum of the previous two, so that we only need to use the following two regressors[8]:

$$\hat{\nabla}_{1i} \equiv \hat{\rho}_i \frac{\hat{q}_i - 1/2}{\hat{q}_i (1-\hat{q}_i)} \quad \text{and} \quad \hat{\nabla}_{2i} \equiv \hat{\rho}_i \frac{\hat{p}_i - 1/2}{\hat{p}_i (1-\hat{p}_i)}.$$

To obtain a double-debiased estimator of $\beta_g = E_w(\rho(x_i) | i \in g)$, we can simply regress $\hat{\rho}_i$ on $\hat{\nabla}_{1i}$, $\hat{\nabla}_{2i}$, and a constant over the sample $i \in g$ with weights w_i. The double-debiased estimator is the coefficient of the constant; we will denote it $\tilde{\rho}_g$ from now on on.

5 Testing the Positive Correlation Property

We implement two ways of testing the positive correlation property: an intersection test and a sorted group approach, in Sections 5.1 and 5.2 respectively. Both methods require that we use cross-fitted predictors of the covariance and correlation; we use the estimates described in Section 3.2 for this purpose.

[8] Using $p = p_{10} + p_{11}$ and $q = p_{01} + p_{11}$.

5.1 The intersection test

It is clearly not feasible to test that the covariance (or correlation) is positive for all possible values of all covariates. However, results in Semenova and Chernozhukov (2021) show that we can use standard inference for their mean values over subgroups of observations.

More precisely, let T denote either covariance or correlation, and consider testing the following hypothesis:

$$C(\bar{h}) \equiv E(T(X)|h(X) = \bar{h}) \ \ \forall \bar{h} \text{ in } B, \tag{2}$$

where h is a function whose values lie in a low-dimensional space, and B is a subset of its range. We could for instance test that the positive correlation property holds on average over all women in rural areas who drive cars that are more than 5 year old.

Semenova and Chernozhukov (2021) derive assumptions under which one can run a sieve regression

$$\hat{T}(x_i) = p(h(x_i))\beta + u_i$$

and use its fitted values $p(h(x_i))\hat{\beta}$, instead of the neural network estimate of $T(x_i)$, to test the multiple hypothesis (2). Note that p applies to the values of h; in the example of the previous paragraph, p could only be a function of gender, whether the insuree lives in a rural area, and whether the car is more than 5 years old.

The statistic T must be double-debiased if needed (that is, we use $\hat{C}$ and $\tilde{\rho}$); it must converge faster than $n^{-1/4}$, which holds under reasonable assumptions on the neural network; and the sieve basis p must expand at the appropriate rate, which implicitly limits the dimensionality of the function h.

The testing procedure simplifies further if we focus on groups of observations. To be more precise, suppose that we define disjoint groups $g_1, \ldots, g_L$ by the values of some of our covariates. The union of the groups could be the whole set of observations, but that is not necessary. For instance, g_1 could have all young men and g_2 all women who have an old car. Our goal is to test the null hypothesis that

$$E(T(X)|X \in g_l) \geq 0 \ \text{ for } \ l = 1, \ldots, L$$

where T is either the covariance or the correlation. In the notation of the previous paragraphs, this amounts to using indicators of the groups as the basis functions $p \circ h$ in the sieve regression; the model is saturated and the sieve basis trivially satisfies the conditions in Semenova and Chernozhukov (2021).

Then for each of our groups g_l, we compute the sample-weighted average predicted covariance $\hat{C}_l$. We apply the double-debiasing procedure described in the previous subsection to obtain $\tilde{\rho}_l$,

the double-debiased, sample-weighted average correlation within this group. We compute the standard error $\hat{\sigma}_l$ of the estimators $\hat{T}_l = \tilde{\rho}_l$ or $\hat{C}_l$ by the usual formula.

If $L = 1$ (for instance, we only want to test the positive correlation property for young men), then we are done: we reject the null hypothesis at the 5% level if $\hat{T}_l + 1.64\hat{\sigma}_l < 0$. If $L > 1$, we want to test that $\min_{l=1,\dots,L} E(T(X)|X \in g_l) > 0$. This is an intersection test; we use the procedure described in Chernozhukov, Lee, and Rosen (2013):

1. we draw a large number of values $(\xi_r)_{r=1,\dots,R}$ from $N(0, I_L)$;
2. for each r, we define $\bar{v}_r = \max_{l=1,\dots,L} \xi_{rl}$; we let k_0 be the γ_n-quantile of the $\bar{v}_r$, with $\gamma_n = 1 - 0.1/\log n$;
3. we let $\hat{L}$ be the set of values of l such that

$$\hat{T}_l \leq \min_{m=1,\dots,L} \left(\hat{T}_m + k_0 \hat{\sigma}_m \right) + 2k_0 \hat{\sigma}_l;$$

4. finally, we let k be the $(1-\alpha)$-quantile of the values

$$\hat{v}_r = \max_{l \in \hat{L}} \xi_{rl}$$

and we reject the hypothesis if $\inf_{l \in \hat{L}} \left(\hat{T}_l + k\hat{\sigma}_l \right) < 0$.

In Sections 5.1.1 and 5.1.2, we will implement the intersection test for the covariance and the correlation coefficient, respectively. To define the groups g_l, we use the age of the car, which is by far the variable with the most predictive content. We split it into quartiles, and we also run tests for all of its modalities.

5.1.1 The results of the intersection test for the covariance

Table 4 gives the results of the intersection tests for the variable "car age" and its split by quartiles (the results of the intersection tests for other variables are provided in Table 13 in the Appendix)[9].

Note that k (the number of standard errors used in the last step of the intersection test) varies noticeably. For the 5% test, it varies from 1.62 (that is, the standard value) to 2.54, which is larger than what a naive normal approximation would suggest. The test statistics show that for any of the partitions into groups, we can reject the hypothesis that the covariances are

[9]The bounds of the confidence intervals are computed as $a = \min_l(\hat{C}_l + k\hat{\sigma}_l)$ and $b = \max_l(\hat{C}_l - k\hat{\sigma}_l)$: these are the values such that we are at the margin of rejecting the hypothesis that $\min_l C_l > a$ and the hypothesis that $\max_l C_l < b$.

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car age	4	0.01	2.71	2.17	-0.0023	rejected	[-0.0023,0.0011]
quartiles		0.05	2.71	1.57	-0.0023	rejected	[-0.0023,0.0012]
		0.10	2.71	1.28	-0.0023	rejected	[-0.0023,0.0012]
Car age	12	0.01	3.12	2.94	-0.0026	rejected	[-0.0025,0.0011]
		0.05	3.12	2.41	-0.0026	rejected	[-0.0026,0.0012]
		0.10	3.12	2.08	-0.0027	rejected	[-0.0026,0.0013]

Table 4: PCP test using the covariances for the "age of the car" variable variable: neural network estimates

all positive at any reasonable level. On the other hand, we can also reject that any of them is smaller (i.e., less negative) than, say, -0.01; this can be seen in the "confidence interval" column.

5.1.2 The results of the intersection test for the correlation

Let us turn to the correlation function. The results in Table 5 are very similar to those for the covariance: once again, the positive correlation property is rejected for all partitions of the sample (the test results for other variables are shown in Table 14 in the Appendix).

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car age	4	0.01	2.71	2.61	-0.0034	rejected	[-0.0034,0.0003]
quartiles		0.05	2.71	2.10	-0.0036	rejected	[-0.0035,0.0004]
		0.10	2.71	1.77	-0.0037	rejected	[-0.0036,0.0005]
Car age	12	0.01	3.12	2.99	-0.0035	rejected	[-0.0035,0.0012]
		0.05	3.12	2.52	-0.0038	rejected	[-0.0037,0.0016]
		0.10	3.12	2.20	-0.0040	rejected	[-0.0038,0.0018]

Table 5: PCP test using the correlation coefficient for the "car age of the car" variable: neural network estimates

Moreover, the confidence intervals confine the (doubly-debiased) correlation to a narrow interval just below zero. This is not at all what Figure 5 (which plots the density of $\hat{\rho}$, not that of the double-debiased version $\tilde{\rho}$) would have suggested; it is a good illustration of the fact that raw predictions from neural networks are not free from bias and noise.

5.2 The sorted groups approach

A recent contribution by Chernozhukov, Demirer, Duflo, and Fernández-Val (2023) adopts a different approach to recover estimators of group averages that have standard asymptotics.

While they define it in the more complex framework of conditional average treatment effects, their method a fortiori applies in our setting. The underlying idea is to split the sample into a main sample and an auxiliary subsample. A machine learning model is trained and optimized on the auxiliary subsample to predict the statistics of interest (here, the probabilities of the four alternatives). The predictors are applied to the observations in the main sample, which are then allocated into groups sorted by the value of, in our case, the predicted covariance or correlation. Chernozhukov, Demirer, Duflo, and Fernández-Val (2023) show that regressing the outcomes y_{jk} observed in the main sample on group indicators gives estimators of the average probabilities that have standard asymptotics and can therefore be used to construct standard tests.

5.2.1 The results of the sorted groups approach for the covariance

As recommended by Chernozhukov, Demirer, Duflo, and Fernández-Val (2023), we use several random splits into main and auxiliary samples, and we report the median test statistics and p-values. We train and select a neural network on the auxiliary subsample exactly as explained in Section 3. Once we have predicted probabilities $\hat{p}_{jk}$ on the main sample we use them to compute the covariance for each observation using the formulæ in Section 4.1. We then sort the observations according to the predicted covariance, and we define four groups $q = 1, 2, 3, 4$, splitting at the quartiles. In each group, we regress y_{jk} on the group indicators to obtain new predicted group-average probabilities $\bar{p}_{jk}(q)$. The results in Chernozhukov, Demirer, Duflo, and Fernández-Val (2023) imply that these $\bar{p}$ statistics have standard asymptotics. This allows us to test the PCP on each group q, and to define confidence intervals for the q-group covariances.

The results are reported in Table 6, where the medians are computed over 100 random splits. While the PCP is not rejected at the 5% level, the confidence interval for the average covariance on the lowest-covariance quartile is very narrow. This is consistent with the results in Tables 4 and 5.

Statistic	Covariance in quartile 1	Estimated standard error	95% confidence interval	Test statistic	p-value
Median over 101 splits	-0.0030	0.0024	$[-0.0076, 0.0016]$	-1.28	0.10

Table 6: Testing for a positive covariance on the first covariance quartile

5.2.2 The results of the sorted groups approach for the correlation

We proceed in exactly the same way for the correlation. Remarkably, the main/auxiliary method used in Chernozhukov, Demirer, Duflo, and Fernández-Val (2023) allows us to circumvent

double-debiasing altogether. As Table 7 shows, we come close to rejecting the PCP for the lowest-correlation quartile.

Statistic	Correlation in quartile 1	Estimated standard error	95% confidence interval	Test statistic	p-value
Median over 101 splits	-0.0304	0.0207	$[-0.0672, 0.0083]$	-1.52	0.06

Table 7: Testing for a positive correlation on the first correlation quartile

The confidence interval shows once more that only very small values of the correlation are consistent with the data.

6 Deep learning vs tree-based methods

For the sake of comparison, in this section we use another popular machine approach: ensemble learning applied to decision trees. To train the ensemble, we consider two alternative learners: bagging and boosting. Bagging applied to feature selection using decision trees is referred to as "random forests", while the boosting method yields gradient-boosted trees. Using these two methods, we estimate the covariance and the correlation functions, we double-debias the correlation, and we test the positive correlation property using the intersection test.

6.1 Model selection

In the machine learning setting, predictions from all types of models can be averaged. This approach has become especially popular with models based on decision trees — ensemble learners that average over a large numbers of shallow decision trees that are trained over random subsamples and features.

We use $N = 500$ ensemble members and 5-fold cross validation with the weighted entropy criterion. We do a grid search over three hyperparameter values: the maximum depth of each decision tree; the minimum number of observations in each leaf; and the maximum number of features that are randomly selected before each split (for the random forest) or the learning rate (for the gradient-boosted tree). To optimize in hyperparameter space, we split the data into training and testing sets, representing 80% and 20% of the data respectively; we select the model that gives the best fit on a test sample. Both procedures report "feature importance" scores; we plot them in Figures 7 and 8.

The best random forest has a maximum depth of 5; a minimum leaf size of 10; and randomization over a maximum of 5 features. Figure 7 shows that the age of the car again is the most important explanatory variable by far. The best gradient-boosted tree has a maximum depth

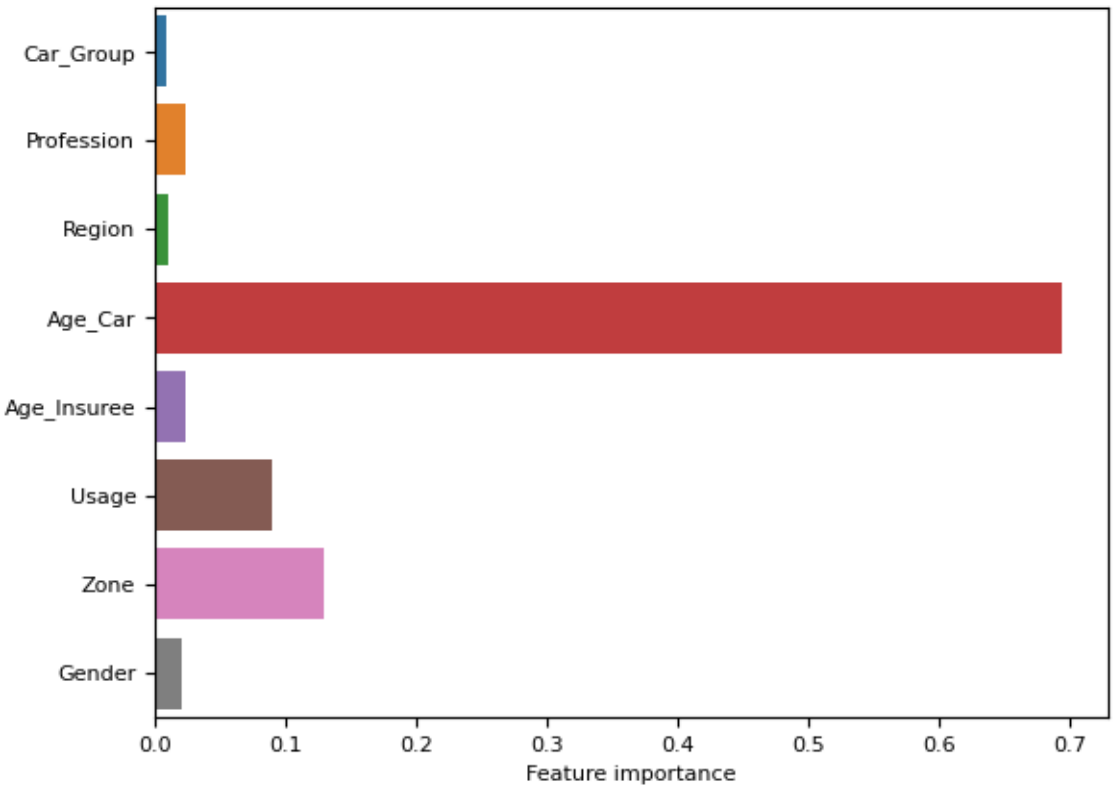


Figure 7: Feature importances for the best random forest

of 4; a minimum leaf size of 20; and a learning rate of 0.1. Figure 8 shows that the importance of the various features for the gradient boosting is very similar to that for our selected random forest.

When computed over the whole sample, the weighted entropy losses of the three methods are 0.386 for the neural network, 0.392 for the gradient-boosted tree, and 0.402 for the random forest. We were surprised that the tree-based learners do not work better on this tabular data. These differences are not very large, however, and they may be quite specific to our dataset.

6.2 Comparison results for the covariance and correlation functions

Table 8 reports our estimates of the covariances and correlation coefficients produced by random forest and gradient boosting methods.

Figure 9 plots the density of the predicted correlations $\hat{\rho}_i$ over the sample under all three of our machine learning methods: deep learning, random forest, and gradient boosting. The

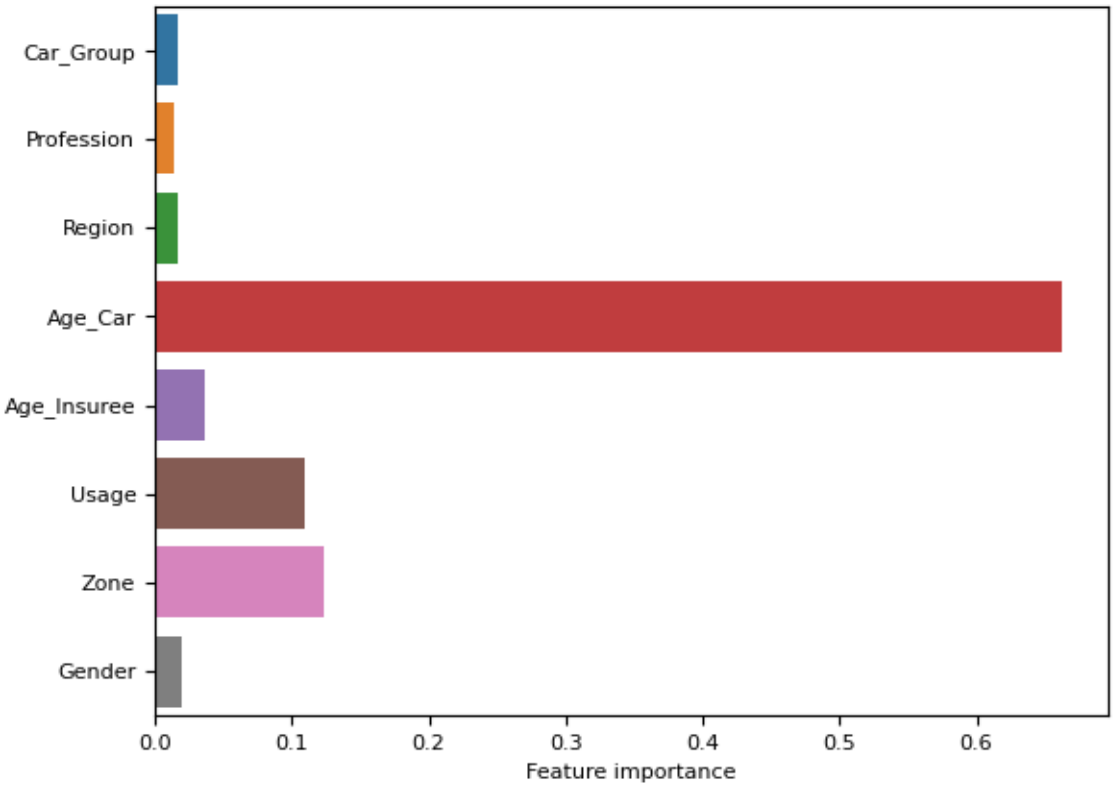


Figure 8: Feature importances for the best gradient boosting method

Name of variable	Random Forest			Gradient Boosting		
	Mean	Dispersion	Range	Mean	Dispersion	Range
Covariance	-0.002	0.004	[-0.028, 0.012]	-0.003	0.009	[-0.066, 0.069]
Correlation	-0.018	0.035	[-0.198, 0.107]	-0.032	0.083	[-0.420, 0.503]

Table 8: Raw random-forest and gradient-boosting estimates for the covariance and correlation

range of values of the correlation coefficient is similar for all three methods: essentially all mass belongs to the $[-0.2, 0.2]$ interval. While the density of $\rho(x)$ obtained from the neural network is bimodal, with two asymmetric peaks in the negative and positive ranges, the densities produced by the ensembles are unimodal. Another important difference is that the gradient-boosted tree produces a much broader range of variation than the other two methods on this data.

Figure 10 (resp. Figure 11) shows the correlation for different groups produced by the random forest method (resp. the gradient boosting method). These plots show some marked differences with the equivalent plot for the neural network (Figure 6), most notably for the important "Age of car" variable.

6.3 The intersection test

We repeated the intersection tests of the positive correlation property with our random forest and our gradient-boosted tree. We focus on the correlation to save space. Table 9 reports the results for the random forest method, and Table 10 gives them for the gradient boosting method (the results for other variables are provided in the Appendix).

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car age	4	0.01	2.80	2.37	-0.0033	rejected	[-0.0033,-0.0003]
quartiles		0.05	2.80	1.63	-0.0034	rejected	[-0.0034,-0.0002]
		0.10	2.80	1.23	-0.0035	rejected	[-0.0034,-0.0001]
Car age	12	0.01	3.18	2.18	-0.0074	rejected	[-0.0073,0.0044]
		0.05	3.18	1.68	-0.0077	rejected	[-0.0075,0.0045]
		0.10	3.18	1.21	-0.0080	rejected	[-0.0077,0.0045]

Table 9: PCP test using the correlation coefficient for the "car age" variable: random forest estimation

Since the lower bounds are negative for all methods, we can reject the hypothesis that the covariances are all positive. The lower bounds given by the random forest and especially the gradient-boosted tree are lower than with the neural network; still, even with the gradient-boosted tree the lower bound is not economically significant from zero. We can safely conclude that the three machine learning methods produce qualitatively similar implications.

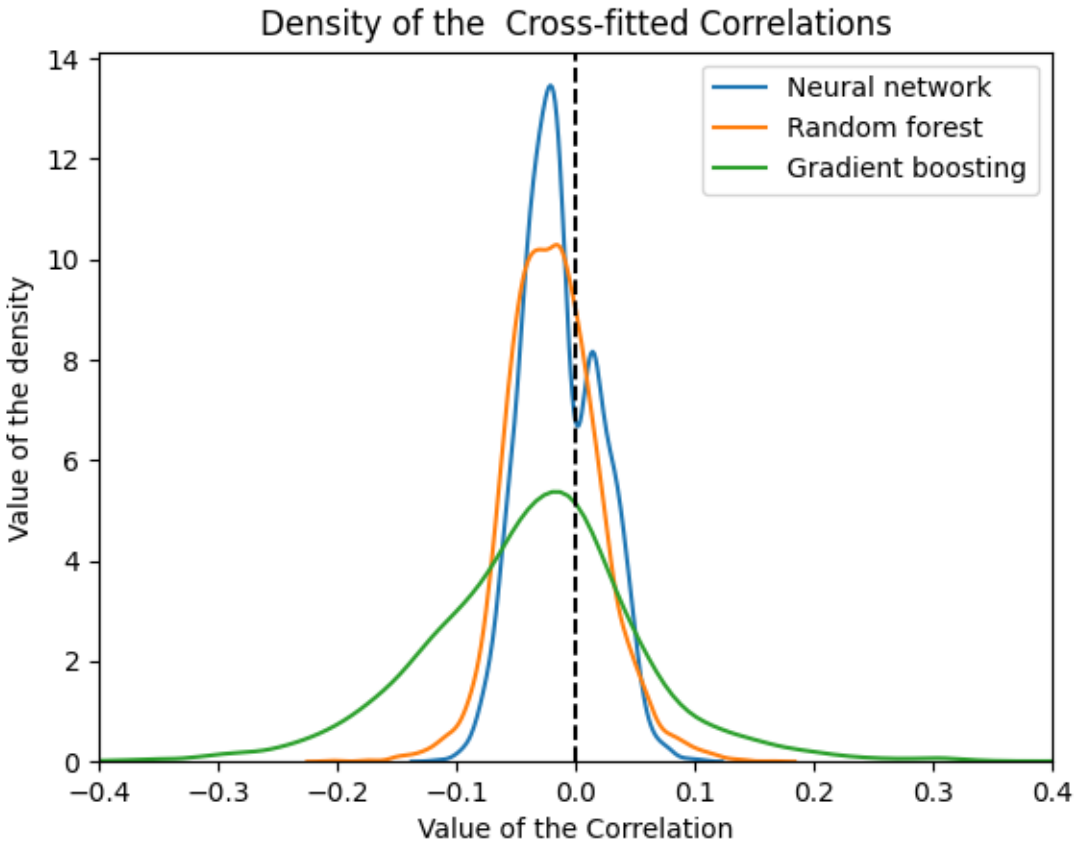


Figure 9: Density of $\hat{\rho}(x)$ for the three machine learners

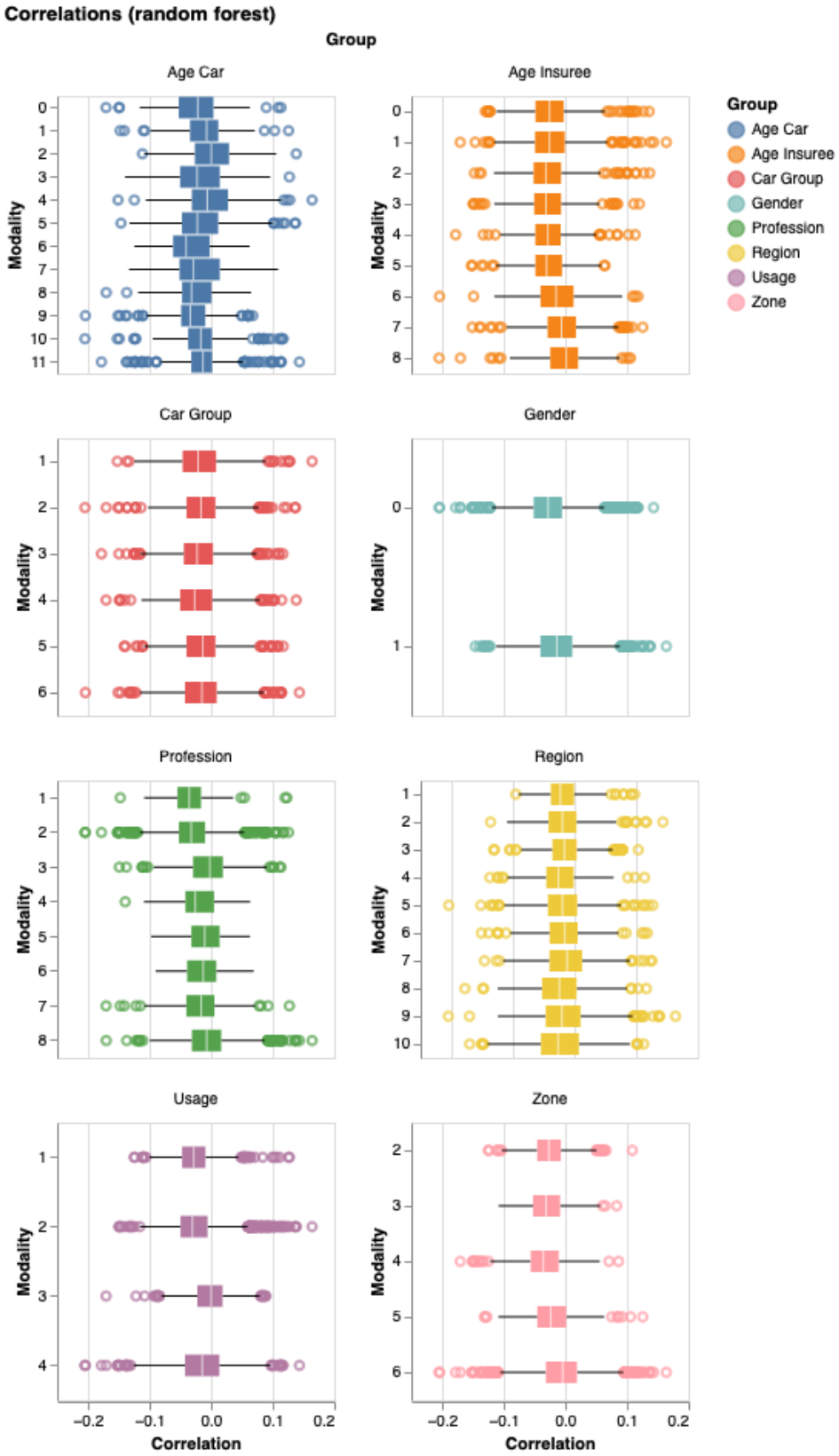


Figure 10: Correlation $\rho(x)$ obtained from the random forest for different groups

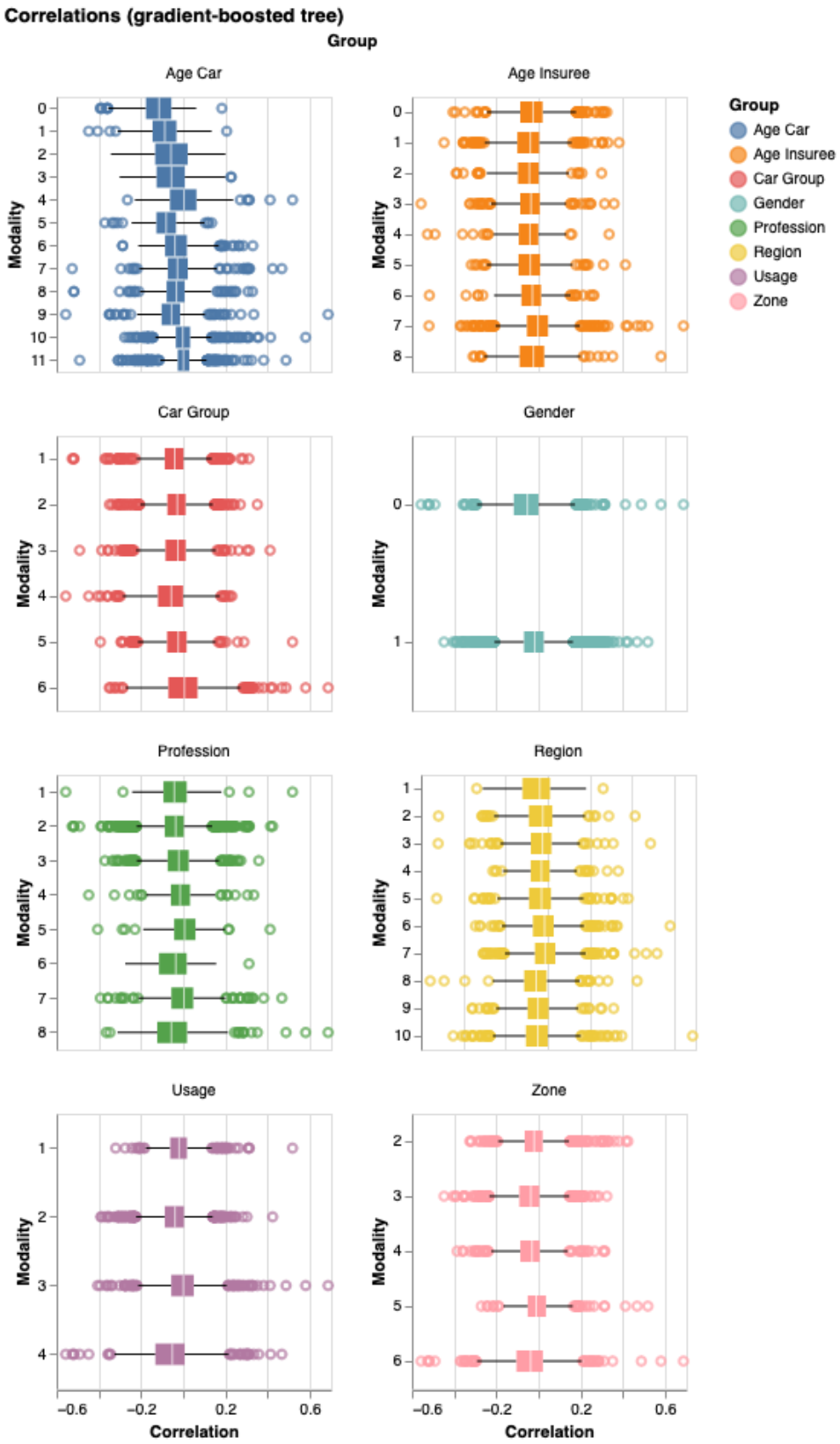


Figure 11: Correlation $\rho(x)$ obtained from gradient boosting for different groups

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car age	4	0.01	2.80	2.67	-0.0159	rejected	[-0.0158,-0.0086]
quartiles		0.05	2.80	1.94	-0.0160	rejected	[-0.0159,-0.0085]
		0.10	2.80	1.64	-0.0160	rejected	[-0.0160,-0.0085]
Car age	12	0.01	3.18	2.75	-0.0262	rejected	[-0.0255,0.0183]
		0.05	3.18	2.13	-0.0278	rejected	[-0.0272,0.0189]
		0.10	3.18	1.85	-0.0286	rejected	[-0.0278,0.0191]

Table 10: PCP test using the correlation coefficient for the "car age" variable: gradient-boosted estimation

6.4 The group-averaged correlations

Finally, Tables 11 and 12 report the group-averaged correlations produced by the three machine learners at two levels of grouping on the "Age of the car" variable: with respectively four and twelve modalities[10]. The great majority of the correlations are negative, and they tend to shrink after double-debiasing (indicated by "DD").

	Neural network		Random forest		Gradient-boosted tree	
Modality	Raw	DD	Raw	DD	Raw	DD
1	0.0152	0.0009 (0.0003)	-0.0106	−0.0014 (0.0002)	-0.0640	−0.0124 (0.0009)
2	-0.0139	−0.0041 (0.0002)	-0.0236	0.0002 (0.0002)	-0.0399	−0.0078 (0.0005)
3	-0.0300	−0.0042 (0.0003)	-0.0232	−0.0037 (0.0002)	-0.0235	−0.0162 (0.0001)
4	-0.0263	−0.0028 (0.0004)	-0.0148	−0.0025 (0.0001)	0.0019	−0.0080 (0.0002)

Table 11: Group-averaged correlations: car age quartiles

Concluding Remarks

Even with the very flexible methods used in our paper, this dataset shows no evidence for the positive correlation property. In addition to this empirical finding, our paper contains a methodological contribution. When we started this project, it was not clear to us that deep learning methods could be applied fruitfully to a non-trivial testing problem on such a (relatively) small dataset. With our sample of just $6,333$ observations, deep learning cannot go very deep: neural networks with at most two hidden layers and a small dropout rate work

[10]For completeness, the Appendix compares the group-averaged correlations produced by the three methods for the other variables.

	Neural network		Random forest		Gradient-boosted tree	
Modality	Raw	DD	Raw	DD	Raw	DD
0	0.0180	0.0033 (0.0007)	-0.0243	−0.0089 (0.0007)	-0.1206	−0.0337 (0.0027)
1	0.0173	0.0018 (0.0006)	-0.0106	−0.0022 (0.0003)	-0.0911	−0.0200 (0.0020)
2	0.0180	0.0013 (0.0007)	0.0013	0.0046 (0.0002)	-0.0549	−0.0085 (0.0020)
3	0.0116	−0.0003 (0.0005)	-0.0169	0.0009 (0.0007)	-0.0561	−0.0102 (0.0019)
4	0.0110	0.0011 (0.0006)	-0.0012	0.0051 (0.0002)	0.0075	0.0205 (0.0009)
5	-0.0012	−0.0029 (0.0003)	-0.0156	0.0032 (0.0004)	-0.0793	−0.0152 (0.0019)
6	-0.0155	−0.0034 (0.0004)	-0.0331	0.0003 (0.0004)	-0.0272	0.0027 (0.0007)
7	-0.0226	−0.0045 (0.0005)	-0.0209	−0.0018 (0.0003)	-0.0199	−0.0096 (0.0005)
8	-0.0276	−0.0051 (0.0005)	-0.0252	−0.0039 (0.0003)	-0.0342	−0.0167 (0.0006)
9	-0.0320	−0.0033 (0.0005)	-0.0302	−0.0030 (0.0003)	-0.0548	−0.0139 (0.0010)
10	-0.0302	−0.0037 (0.0005)	-0.0175	−0.0023 (0.0002)	0.0034	−0.0078 (0.0002)
11	-0.0263	−0.0028 (0.0004)	-0.0148	−0.0025 (0.0001)	0.0019	−0.0080 (0.0002)

Table 12: Group-averaged correlations: car age

best. Still, they deploy many more parameters than any econometric model. Our neural network in fact predict the choice of coverage much better than parametric procedures, and they also outperform them on claim occurrence. Double-debiasing, or sorted groups, give us consistent and asymptotically normal estimates that can be used in standard test procedures. Our overall conclusion is that deep learning can provide both robust and useful conclusions even for relatively small-scale applications. Methods based on decision trees—random forests and gradient boosting—also perform well, and give very similar results. In further work, we plan to apply our three-step method to larger samples of insurees, and to explore the value of reducing the number of covariates to those that seem to have the largest effect.

References

Abadi, M., et al. (2016): "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *CoRR*, abs/1603.04467.

Chernozhukov, V., M. Demirer, E. Duflo, and I. Fernández-Val (2023): "Generic Machine Learning Inference On Heterogenous Treatment Effects in Randomized Experiments, With An Application To Immunization In India," mimeo MIT.

Chernozhukov, V., et al. (2018): "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, 21(1), C1–C68.

Chernozhukov, V., S. Lee, and A. M. Rosen (2013): "Intersection Bounds: Estimation and Inference," *Econometrica*, 81(2), 667–737.

Chiappori, P., B. Jullien, B. Salanié, and F. Salanié (2006): "Asymmetric information in insurance: general testable implications," *The RAND Journal of Economics*, 37(4), 783–798.

Chiappori, P., and B. Salanié (2000): "Testing for Asymmetric Information in Insurance Markets," *Journal of Political Economy*, 108(1), 56–78.

——— (2013): "Asymmetric Information in Insurance Markets: Predictions and Tests," in *Handbook of Insurance*, ed. by G. Dionne, pp. 397–422. Springer New York.

Chollet, F. (2021): *Deep Learning with Python, Second Edition*. Manning.

Dionne, G., C. Gouriéroux, and C. Vanasse (2001): "Testing for evidence of adverse selection in the automobile insurance market: A comment," *Journal of Political Economy*, 109, 444–453.

——— (2006): "Informational content of household decisions with applications to insurance under asymmetric information," in *Competitive Failures in Insurance Markets: Theory and Policy Implications*, ed. by P. Chiappori, and C. Gollier, CESifo Seminar Series, pp. 159–184. MIT Press, Cambridge, MA:.

Kim, H., D. Kim, S. Im, and J. W. Hardin (2009): "Evidence of Asymmetrie Information in the Automobile Insurance Market: Dichotomous Versus Multinomial Measurement of Insurance Coverage," *Journal of Risk and Insurance*, 76, 343–366.

Kingma, D. P., and J. Ba (2017): "Adam: A Method for Stochastic Optimization," Discussion paper, arXiv 1412.6980.

Rothschild, M., and J. Stiglitz (1976): "Equilibrium in Competitive Insurance Markets: An Essay on the Economics of Imperfect Information," *The Quarterly Journal of Economics*, 90(4), 629–649.

Semenova, V., and V. Chernozhukov (2021): "Debiased Machine Learning of Conditional Average Treatment Effects and Other Causal Functions," *Econometrics Journal*, 24, 264–289.

Spindler, M. (2014): "Econometric Methods for Testing for Asymmetric Information: A Comparison of Parametric and Nonparametric Methods with an Application to Hospital Daily Benefits," *The Geneva Risk and Insurance Review*, 39, 254–266.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014): "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, 15, 1929–1958.

Su, L., and M. Spindler (2013): "Nonparametric Testing for Asymmetric Information," *Journal of Business and Economic Statistics*, 31, 208–225.

Appendix: additional results

Tables 13 and 14 show the results of intersection tests for all groups of variables except the age of the car (for which we reported the results in Tables 4 and 5 in the main text), using our neural network estimates. Table 13 focuses on covariances and Table 14 on correlations.

Tables 15 and 16 shows the results of the intersection test for group-averaged correlation functions for the random forest and the gradient-boosted tree. Finally, Tables 17 to 23 give the values of the group-averaged correlations for all modalities of other variables than the age of the car, for the three machine-learning methods[11].

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car group	6	0.01	2.96	2.85	-0.0014	rejected	[-0.0014,-0.0006]
		0.05	2.96	2.31	-0.0015	rejected	[-0.0014,-0.0006]
		0.10	2.96	2.00	-0.0015	rejected	[-0.0015,-0.0005]
Insuree age	9	0.01	3.11	3.07	-0.0010	rejected	[-0.0009,-0.0005]
		0.05	3.11	2.58	-0.0010	rejected	[-0.0010,-0.0005]
		0.10	3.11	2.27	-0.0011	rejected	[-0.0010,-0.0004]
Gender	2	0.01	2.60	2.61	-0.0009	rejected	[-0.0009,-0.0008]
		0.05	2.60	2.00	-0.0009	rejected	[-0.0009,-0.0008]
		0.10	2.60	1.65	-0.0010	rejected	[-0.0009,-0.0008]
Zone	5	0.01	2.89	2.80	-0.0014	rejected	[-0.0013,-0.0004]
		0.05	2.89	2.26	-0.0014	rejected	[-0.0014,-0.0004]
		0.10	2.89	1.99	-0.0014	rejected	[-0.0014,-0.0003]
Usage	4	0.01	2.71	2.61	-0.0011	rejected	[-0.0011,-0.0007]
		0.05	2.71	2.02	-0.0012	rejected	[-0.0012,-0.0006]
		0.10	2.71	1.70	-0.0012	rejected	[-0.0012,-0.0006]
Profession	8	0.01	3.03	2.97	-0.0012	rejected	[-0.0012,-0.0005]
		0.05	3.03	2.52	-0.0013	rejected	[-0.0012,-0.0005]
		0.10	3.03	2.20	-0.0013	rejected	[-0.0013,-0.0005]
Region	10	0.01	3.12	3.14	-0.0008	rejected	[-0.0012,-0.0008]
		0.05	3.12	2.62	-0.0009	rejected	[-0.0011,-0.0009]
		0.10	3.12	2.38	-0.0010	rejected	[-0.0011,-0.0009]

Table 13: PCP test using the covariances for variables other than the age of the car: neural network estimates

[11] Results for the covariances are available upon request.

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car group	6	0.01	2.96	2.85	-0.0070	rejected	[-0.0069,-0.0044]
		0.05	2.96	2.31	-0.0071	rejected	[-0.0071,-0.0043]
		0.10	2.96	2.00	-0.0072	rejected	[-0.0071,-0.0042]
Insuree age	9	0.01	3.11	2.97	-0.0067	rejected	[-0.0067,-0.0037]
		0.05	3.11	2.51	-0.0069	rejected	[-0.0069,-0.0036]
		0.10	3.11	2.23	-0.0071	rejected	[-0.0069,-0.0035]
Gender	2	0.01	2.60	2.61	-0.0054	rejected	[-0.0062,-0.0054]
		0.05	2.60	2.00	-0.0055	rejected	[-0.0061,-0.0055]
		0.10	2.60	1.65	-0.0056	rejected	[-0.0061,-0.0055]
Zone	5	0.01	2.89	2.72	-0.0068	rejected	[-0.0067,-0.0040]
		0.05	2.89	2.16	-0.0070	rejected	[-0.0069,-0.0039]
		0.10	2.89	1.80	-0.0071	rejected	[-0.0070,-0.0039]
Usage	4	0.01	2.71	2.80	-0.0048	rejected	[-0.0057,-0.0048]
		0.05	2.71	2.25	-0.0050	rejected	[-0.0056,-0.0049]
		0.10	2.71	1.97	-0.0051	rejected	[-0.0056,-0.0050]
Profession	8	0.01	3.03	2.99	-0.0062	rejected	[-0.0060,-0.0044]
		0.05	3.03	2.54	-0.0067	rejected	[-0.0065,-0.0044]
		0.10	3.03	2.24	-0.0070	rejected	[-0.0067,-0.0043]
Region	10	0.01	3.12	3.14	-0.0065	rejected	[-0.0064,-0.0057]
		0.05	3.12	2.62	-0.0068	rejected	[-0.0067,-0.0056]
		0.10	3.12	2.38	-0.0069	rejected	[-0.0068,-0.0055]

Table 14: PCP test using the correlation coefficient for variables other than the age of the car: neural network estimates

Group	Modalities	Test level	k_0	k	Test statistic	PCP	CI lowerCI upper
Car group	6	0.01	2.89	2.80	-0.0035	rejected	[-0.0035, -0.0003]
		0.05	2.89	2.13	-0.0036	rejected	[-0.0036, -0.0002]
		0.10	2.89	1.89	-0.0037	rejected	[-0.0036, -0.0002]
Insuree age	9	0.01	3.09	2.93	-0.0066	rejected	[-0.0065, 0.0031]
		0.05	3.09	2.31	-0.0067	rejected	[-0.0066, 0.0032]
		0.10	3.09	2.03	-0.0068	rejected	[-0.0067, 0.0032]
Gender	2	0.01	2.45	2.32	-0.0041	rejected	[-0.0040, -0.0014]
		0.05	2.45	1.64	-0.0043	rejected	[-0.0042, -0.0013]
		0.10	2.45	1.24	-0.0044	rejected	[-0.0043, -0.0013]
Zone	5	0.01	2.80	2.80	-0.0017	rejected	[-0.0016, 0.0031]
		0.05	2.80	2.20	-0.0017	rejected	[-0.0017, 0.0032]
		0.10	2.80	1.92	-0.0018	rejected	[-0.0017, 0.0032]
Usage	4	0.01	2.80	2.36	-0.0060	rejected	[-0.0060, 0.0001]
		0.05	2.80	1.68	-0.0062	rejected	[-0.0061, 0.0001]
		0.10	2.80	1.28	-0.0063	rejected	[-0.0062, 0.0001]
Profession	8	0.01	3.03	2.80	-0.0039	rejected	[-0.0039, 0.0013]
		0.05	3.03	2.12	-0.0041	rejected	[-0.0040, 0.0014]
		0.10	3.03	1.83	-0.0041	rejected	[-0.0041, 0.0014]
Region	10	0.01	3.09	2.88	-0.0042	rejected	[-0.0041, -0.0010]
		0.05	3.09	2.43	-0.0044	rejected	[-0.0043, -0.0009]
		0.10	3.09	2.13	-0.0046	rejected	[-0.0044, -0.0008]

Table 15: PCP test using the correlation coefficient for other variables than the age of the car: random forest estimation

Group	Modalities	Test level	k_0	k	Test statistic	PCP	Confidence interval
Car group	6	0.01	2.89	2.75	-0.0149	rejected	[-0.0146, -0.0084]
		0.05	2.89	2.09	-0.0155	rejected	[-0.0153, -0.0082]
		0.10	2.89	1.82	-0.0158	rejected	[-0.0155, -0.0081]
Insuree age	9	0.01	3.09	2.99	-0.0183	rejected	[-0.0182, -0.0030]
		0.05	3.09	2.41	-0.0186	rejected	[-0.0185, -0.0027]
		0.10	3.09	2.11	-0.0188	rejected	[-0.0186, -0.0025]
Gender	2	0.01	2.45	2.32	-0.0155	rejected	[-0.0153, -0.0084]
		0.05	2.45	1.64	-0.0159	rejected	[-0.0157, -0.0083]
		0.10	2.45	1.24	-0.0162	rejected	[-0.0159, -0.0082]
Zone	5	0.01	2.80	2.25	-0.0165	rejected	[-0.0163, -0.0059]
		0.05	2.80	1.63	-0.0168	rejected	[-0.0166, -0.0054]
		0.10	2.80	1.27	-0.0169	rejected	[-0.0168, -0.0052]
Usage	4	0.01	2.80	2.36	-0.0195	rejected	[-0.0193, 0.0006]
		0.05	2.80	1.68	-0.0201	rejected	[-0.0198, 0.0009]
		0.10	2.80	1.28	-0.0206	rejected	[-0.0201, 0.0011]
Profession	8	0.01	3.03	2.86	-0.0137	rejected	[-0.0136, -0.0059]
		0.05	3.03	2.36	-0.0139	rejected	[-0.0138, -0.0056]
		0.10	3.03	2.11	-0.0140	rejected	[-0.0139, -0.0055]
Region	10	0.01	3.09	2.99	-0.0166	rejected	[-0.0162, -0.0033]
		0.05	3.09	2.52	-0.0172	rejected	[-0.0169, -0.0031]
		0.10	3.09	2.23	-0.0176	rejected	[-0.0172, -0.0030]

Table 16: PCP test using the correlation coefficient for for other variables than the age of the car: gradient-boosted tree estimation

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
1	-0.0223	−0.0053 (0.0004)	-0.0183	−0.0010 (0.0003)	-0.0432	−0.0139 (0.0008)
2	-0.0123	−0.0058 (0.0003)	-0.0151	0.0002 (0.0002)	-0.0317	−0.0070 (0.0005)
3	-0.0112	−0.0048 (0.0003)	-0.0209	−0.0033 (0.0003)	-0.0371	−0.0102 (0.0005)
4	-0.0024	−0.0036 (0.0003)	-0.0227	−0.0033 (0.0003)	-0.0613	−0.0174 (0.0009)
5	-0.0143	−0.0075 (0.0003)	-0.0142	−0.0036 (0.0002)	-0.0330	−0.0132 (0.0006)
6	-0.0155	−0.0079 (0.0003)	-0.0158	−0.0040 (0.0002)	0.0064	−0.0062 (0.0011)

Table 17: Group-averaged correlations: car group

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
0	-0.0079	−0.0058 (0.0002)	-0.0271	−0.0048 (0.0003)	-0.0580	−0.0169 (0.0006)
1	-0.0152	−0.0058 (0.0002)	-0.0130	−0.0012 (0.0000)	-0.0208	−0.0079 (0.0002)

Table 18: Group-averaged correlations: gender

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
0	-0.0138	−0.0068 (0.0003)	-0.0237	−0.0056 (0.0003)	-0.0306	−0.0125 (0.0005)
1	-0.0150	−0.0075 (0.0003)	-0.0238	−0.0073 (0.0003)	-0.0481	−0.0201 (0.0006)
2	-0.0181	−0.0080 (0.0004)	-0.0273	−0.0049 (0.0005)	-0.0504	−0.0175 (0.0008)
3	-0.0176	−0.0061 (0.0006)	-0.0284	−0.0037 (0.0004)	-0.0414	−0.0128 (0.0010)
4	-0.0191	−0.0073 (0.0007)	-0.0259	−0.0033 (0.0006)	-0.0511	−0.0135 (0.0016)
5	-0.0172	−0.0076 (0.0007)	-0.0284	−0.0029 (0.0006)	-0.0420	−0.0138 (0.0012)
6	-0.0130	−0.0069 (0.0005)	-0.0129	−0.0002 (0.0004)	-0.0336	−0.0138 (0.0011)
7	-0.0076	−0.0045 (0.0003)	-0.0048	0.0011 (0.0001)	-0.0049	−0.0014 (0.0007)
8	-0.0016	−0.0026 (0.0004)	-0.0012	0.0034 (0.0002)	-0.0289	−0.0097 (0.0007)

Table 19: Group-averaged correlations: insuree age

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
1	-0.0149	−0.0070 (0.0008)	-0.0310	−0.0026 (0.0007)	-0.0359	−0.0057 (0.0022)
2	-0.0195	−0.0064 (0.0002)	-0.0305	−0.0045 (0.0002)	-0.0438	−0.0147 (0.0004)
3	-0.0038	−0.0037 (0.0002)	-0.0038	0.0013 (0.0001)	-0.0245	−0.0045 (0.0004)
4	-0.0119	−0.0057 (0.0009)	-0.0189	−0.0024 (0.0002)	-0.0126	−0.0074 (0.0016)
5	-0.0166	−0.0097 (0.0012)	-0.0121	−0.0013 (0.0003)	0.0042	−0.0017 (0.0033)
6	-0.0014	−0.0030 (0.0010)	-0.0146	0.0016 (0.0004)	-0.0438	−0.0041 (0.0024)
7	-0.0172	−0.0040 (0.0004)	-0.0179	−0.0022 (0.0001)	-0.0026	−0.0046 (0.0009)
8	-0.0035	−0.0055 (0.0003)	-0.0048	0.0018 (0.0002)	-0.0483	−0.0115 (0.0011)

Table 20: Group-averaged correlations: profession

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
1	-0.0089	−0.0048 (0.0005)	-0.0194	−0.0023 (0.0004)	-0.0515	−0.0206 (0.0013)
2	-0.0134	−0.0074 (0.0004)	-0.0176	−0.0028 (0.0002)	-0.0361	−0.0158 (0.0009)
3	-0.0135	−0.0045 (0.0005)	-0.0154	−0.0007 (0.0003)	-0.0339	−0.0187 (0.0011)
4	-0.0102	−0.0054 (0.0005)	-0.0233	−0.0057 (0.0005)	-0.0343	−0.0118 (0.0008)
5	-0.0122	−0.0059 (0.0003)	-0.0190	−0.0037 (0.0003)	-0.0343	−0.0148 (0.0006)
6	-0.0159	−0.0060 (0.0003)	-0.0158	−0.0020 (0.0003)	-0.0153	−0.0059 (0.0005)
7	-0.0130	−0.0047 (0.0003)	-0.0124	−0.0004 (0.0002)	-0.0113	−0.0021 (0.0004)
8	-0.0123	−0.0081 (0.0005)	-0.0232	−0.0033 (0.0005)	-0.0554	−0.0093 (0.0013)
9	-0.0119	−0.0045 (0.0005)	-0.0158	−0.0009 (0.0003)	-0.0485	−0.0143 (0.0011)
10	-0.0108	−0.0073 (0.0004)	-0.0236	−0.0041 (0.0004)	-0.0505	−0.0126 (0.0010)

Table 21: Group-averaged correlations: region

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
1	-0.0176	−0.0059 (0.0004)	-0.0263	−0.0009 (0.0002)	-0.0183	−0.0055 (0.0004)
2	-0.0165	−0.0052 (0.0002)	-0.0252	−0.0001 (0.0001)	-0.0427	−0.0070 (0.0004)
3	-0.0072	−0.0054 (0.0002)	-0.0026	0.0001 (0.0000)	-0.0032	0.0020 (0.0006)
4	-0.0070	−0.0056 (0.0003)	-0.0159	−0.0067 (0.0003)	-0.0590	−0.0218 (0.0010)

Table 22: Group-averaged correlations: usage

Modality	Neural network		Random forest		Gradient-boosted tree	
	Raw	DD	Raw	DD	Raw	DD
2	-0.0164	−0.0044 (0.0002)	-0.0245	−0.0020 (0.0001)	-0.0196	−0.0105 (0.0003)
3	-0.0213	−0.0077 (0.0003)	-0.0291	−0.0021 (0.0002)	-0.0449	−0.0175 (0.0005)
4	-0.0231	−0.0047 (0.0005)	-0.0346	−0.0018 (0.0003)	-0.0410	−0.0110 (0.0006)
5	-0.0206	−0.0037 (0.0006)	-0.0212	−0.0006 (0.0003)	-0.0003	−0.0041 (0.0006)
6	-0.0025	−0.0035 (0.0002)	-0.0054	0.0034 (0.0001)	-0.0415	−0.0117 (0.0005)

Table 23: Group-averaged correlations: zone