UNIVERSITA' DEGLI STUDI DI PADOVA

DIPARTIMENTO DI SCIENZE ECONOMICHE ED AZIENDALI
"M. FANNO"

CORSO DI LAUREA MAGISTRALE IN
ECONOMICS AND FINANCE

TESI DI LAUREA

"THE APPLICATION OF MACHINE LEARNING METHODS IN ECONOMICS AND ECONOMETRICS"

RELATORE:

CH.MO PROF. LUCA NUNZIATA

LAUREANDA: SARA BORTOLOTTI

MATRICOLA N. 1228006

ANNO ACCADEMICO 2021 – 2022

Dichiaro di aver preso visione del "Regolamento antiplagio" approvato dal Consiglio del Dipartimento di Scienze Economiche e Aziendali e, consapevole delle conseguenze derivanti da dichiarazioni mendaci, dichiaro che il presente lavoro non è già stato sottoposto, in tutto o in parte, per il conseguimento di un titolo accademico in altre Università italiane o straniere. Dichiaro inoltre che tutte le fonti utilizzate per la realizzazione del presente lavoro, inclusi i materiali digitali, sono state correttamente citate nel corpo del testo e nella sezione 'Riferimenti bibliografici'.

I hereby declare that I have read and understood the "Anti-plagiarism rules and regulations" approved by the Council of the Department of Economics and Management and I am aware of the consequences of making false statements. I declare that this piece of work has not been previously submitted – either fully or partially – for fulfilling the requirements of an academic degree, whether in Italy or abroad. Furthermore, I declare that the references used for this work – including the digital materials – have been appropriately cited and acknowledged in the text and in the section 'References'.

Firma (signature) ……

Contents

Abstract

The contribution of the thesis is to compute a survey of literature showing the main Machine Learning algorithm, how they worked and how they can be classified.
The main classification describes are Supervised and Unsupervised Machine Learning, parametrized and non-parametrized tests.
Then different regularizer depending on the function class.
Subsequently, specification of different regression, in particular Lasso Regression, Ridge Regression, Elastic Net, Regression trees and finally Neural Network.

The work continues by pointing to the best known techniques for improving the performance of the algorithm, SGD, Boosting, Bootstrap, Bagging, Bumping, Orthogonalization, Cross-validation.

Following the table of contents, there are the specification of advantages and disadvantages coming from both Machine Learning and traditional methods, as the OLS method.

Lastly, an examination of different real cases in which the algorithms of Machine Learning are applied.
The main area that are selected: Poverty, Banking and Finance and Politics and Policy.

The thesis provides an overview of Machine Learning and how it can be applied in economics and econometrics, drawing tangible cases.

1 Introduction

The present thesis focuses on Machine Learning methods, the enhancement performance techniques and the application of those methods to a real cases.

The modern area of using statistics and to build things and to make things better, is that we have figured out that experiments and experimental design are a much more interesting science than you thought.
Experimentation was a science with many unanswered questions, how to share control groups across many different treatments, how to deal with different types of persistence, how to design experiments when there is interactions among the units in the experiments.
Those were all that need immediate attention.
Another important issue is, how to design an experiment having a relatively small number of units, for example cities or schools.
How plan an experiment to not release treatment at the same time everywhere to optimized for statistical power?
Those are all questions and topics that have lots of open questions.

Now Economists are sort of Engineer embedded in a system.
It is possible to design much more complex experiments at large scale and this is true and obvious in Search Engines but it is also going to be really important for schools, governments, for healthcare because in the end we are going to be digitally interacting with people in all of these different settings.

If the Economist is not a sort of part of Engineering team, getting the working on how to set things up to do good experiments, we will really miss out a lot of opportunities.

Those are motivations about the discussion structured in this thesis.

Indeed, the objective of this thesis is to understand the mechanism of Machine Learning Algorithms and how these methods can be applied to real life implementation.

The thesis is based on a review of existing literature and economic analysis performed and the consequences of their introduction and/or variation.

The review is structured in three sections.

In the first section, there is an in-depth description of the main Machine Learning algorithms, specifying some categorisations, such as supervised machine learning, parametric and non-parametric tests and different types of regressions.
Next, some techniques for improving the performance of algorithm estimates are defined.

In the second section, the main features of Machine Learning are defined and compared with those of traditional methods (understood as OLS methods). The main advantages and disadvantages of both methods are also explained.

In the last session, three different real-world cases in which Machine Learning algorithms are applied, are explained.
The areas dedicated to these applications are Poverty, Banking and Finance and Politics and Policy.
Within each area, several papers from recent literature are used to describe real cases and how various models are modified once different algorithms are applied.

The thesis closes with a brief discussion about the main conclusion and findings.

2 Machine Learning Methods

2.1 How machine learning works

Machine intelligence and Artificial intelligence are amazing futuristic themes where will allow us to communicate with robots for example.
But if we think about nowadays there are already available applications: phones that do image recognition or automatic translation.
Probably when we think about Machine Learning we expect to talk about how we can program a computer and explain to a computer how to be intelligent or whether a certain pixel map is a face or not a face.
The process entails taking training data and inducing a relationship between x variable, like the pixels in the image case, to some outcome variable like: is this a face or not ($\hat{y}$).

$$\underbrace{\widehat{y}}_{face} = \hat{f}\left(\overbrace{\tilde{x}}^{pixels}\right)^{1}$$

(1)

It seems very familiar to the logic of statistics models.
The real contribution that Machine Learning gives, with respect to the classical statistics, is that it works with very high dimensional data and it uses complex functional forms.

Machine learning does not just provide a better answer or a different answer to the same question but the difference between Machine Learning and many classical econometrics tools is that the question itself is different.

The question of Machine Learning, in particular, supervised Machine Learning, is to answer to a prediction question about finding a good predictor, hence solving a $\hat{y}$ problem while many tools (OLS estimator for example) are about estimation, thus finding a parameter value $\hat{\beta}$.
This means that in Machine Learning the prediction is the key.

[1] From MLESI21: Sendhil Mullainathan & Jann Spiess

From (Mullainathan and Spiess, 2017) it is feasible to understand the prediction problem set up.

Given:

- Training data set $(y_1, x_1), .., (y_n, x_n)$, assumes *i.i.d*
- Loss function $\ell(\hat{y}, y)$

Authors define that the goal is a prediction function $\hat{f}$ with low average loss ("risk")

$$L(\hat{f}) = E_{y,x}\left[\ell\left(\hat{f}(x), y\right)\right]$$

(2)

Where (y, x) distributed same as training.

In order to explain the problem set up, authors defines a different vocabulary with respect the one used in Econometrics.
In particular, the variable y that in Econometrics it is defined as outcome variable, in Machine Learning it is specified as Label.
The x variable, instead, in Econometrics it is determined as Covariate, while in the Machine Learning is specified as Feature.

The purpose of the process concerns the prediction of the variable y from the variable x, in a way that the average loss between the predicted y and the true y is as low as possible.

It is pointed out that, since the environmental is similar to the classical standard regression, the process can be the same.

In particular:

$$E\left(\hat{f}(x) - y\right)^2 \quad [2]$$

(3)

Normally, the goal is to minimized the mean squared error out of sample, meaning to find a function that predicts the y outcome.

---

[2] From MLESI21: Sendhil Mullainathan & Jann Spiess

The typical approach is to choose some function space, for simplicity linear function,

$$\hat{f}(x) = \hat{\beta}'x = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_j \,^{3}$$

(4)

Where, on the training data, solves for the function that minimizes in-sample fit, rather than finding a parameter that minimized the true out of sample error, according to the same criterion.

$$\min_{\hat{\beta}} E(y - \hat{\beta}'x)^2 \rightarrow \min_{\hat{\beta}} \frac{1}{n}\sum_{i=1}^{n} \left(y - \hat{\beta}'x_i\right)^2 \,^{4}$$

(5)

This can be done by replacing the expectation with what happens in the sample by its sample analogue.

In particular, in the setting of standard regression, it means looking for the B.L.U.E estimator, hence the Best Linear Unbiased Estimator.
This estimator is the best among unbiased estimator, with respect to the coefficient β and with error characterized by some specific structure, with the lowest variance and therefore also the lowest mean squared error among all unbiased estimators.

In the setting of Machine Learning, the aim is to find a good prediction.
More specifically, the purpose is to find generically estimators that are better at predicting, when for example errors are normal and when there are three features, in Machine Learning space, as metaphor for being high dimensional.

For all these reasons, the functional form has to be flexible with the aim of capture the relationship between the y variables and the x variables in ways that simple econometrics model may not, but also to define a solution.
Focusing on the flexible functional forms, the intention is to control the Overfitting.

[3] From MLESI21: Sendhil Mullainathan & Jann Spiess
[4] From MLESI21: Sendhil Mullainathan & Jann Spiess

Overfitting, as (D1Etterich, 1995) address, accounts for the fact that if there is too much work on finding the best fit to the training data, there is also the risk that the noise will fit in the data by memorizing various characteristics of the training data as opposed to finding a general predictive rule.

Following the literature of (Mullainathan and Spiess, 2017), the solution is defined as regularization, that is reducing the complexity of the functional form.
As (Gammerman, Vovk and Vapnik, 2013, p. 9) writes, "Regularization theory was one of the first signs of the existence of intelligent inference.".

From (Athey and Imbens, 2019) rather than running OLS,

$$\min_{\hat{\beta}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{\beta}' x_i\right)^2 \tag{6}$$

as with the previous logic, it is possible to run OLS with constraint

$$\min_{\hat{\beta}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{\beta}' x_i\right)^2 \; s.t. \left|\left|\hat{\beta}\right|\right| \leq c \tag{7}$$

reducing the complexity using, for example, norm or pseudo-norm in order to bounce the sophistication.

For the purpose, literature defined the $l1$ norm, which is not a classical norm but it counts the non-zero coefficients.
Instead of fitting a linear regression with all coefficients, it fits the linear regression only for a set of number of coefficients, which in this example it is identified with c.

As (Mullainathan and Spiess, 2017) illustrates, the problem of minimization is not generally feasible to solve.

An alternative solution is the use of LASSO regression, which replaces the constraint on the number of non-zero coefficients by constrained on the size of coefficients.

$$\left|\left|\hat{\beta}\right|\right|_1 = \sum_{j=1}^{k}\left|\hat{\beta}_j\right| \ ^5$$

(8)

The minimization problem evolves in Lagrange optimization problem:

$$\min_{\hat{\beta}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}' x_i\right)^2 + \lambda\sum_{j=1}^{k}\left|\hat{\beta}_j\right|$$

(9)

Specifically, the difference between the realized y and the predicted y subject to $l1$ norm.
This second term has another element which is λ; this is interpretable as a cost (also called penalty) proportional to the sum of the absolute values of the parameters for the complex functional form.
The λ parameter is retrieved from cross-validation.

Cross validation, as (Mullainathan and Spiess, 2017) writes, means: "randomly partition the sample into equally sized subsamples ("folds"). The estimation process then involves successively holding out one of the folds for evaluation while fitting the prediction function for a range of regularization parameters on all remaining folds. Finally, we pick the parameter with the best estimated average performance".

Repeating this it is possible to leave out data points and fit only on part of the sample and then evaluate on the leftover samples.

In the end, because the priority is prediction it is plausible to evaluate how well prediction algorithm does by taking a random split between training data and test data, fitting function on training data and then evaluating on test data.

---

[5] From *Journal of Economic Perspectives—Volume 31, Number 2—Spring 2017, p. 93, Table 2*

This can be done for example for different values of λ, obtaining an unbiased estimates of the performance, at different costs, depending on different values we choose.
Then the data can be used to decide how much complexity to allow.
All of this is possible because target is prediction, inherently observable.

2.2 Machine Learning Algorithm

2.2.1 Supervised and Unsupervised Learning

In Machine Learning, one of the first classifications outlined in the literature is the separation of learning tasks according to the type of interaction between the learner and the environment.
There are two basic approaches: supervised and unsupervised learning.
The main difference is that one uses labelled data to predict outcomes, while the other does not.

More specifically, (Delua, 2021) designates, Supervised Learning uses labels for input and output data, hence using label datasets.
The algorithm "learns" from the training dataset by repeatedly making predictions about the data, adjusting the correct answers, and simultaneously measuring the accuracy of the predictions.

(Shalev-Shwartz and Ben-David, 2013) provide an example to understand the logic behind the algorithm.
Consideration of two learning tasks: spam detection and anomaly detection.
In the first task, the authors considers a situation in which a learner receives a training email. This email is labelled spam/non-spam.
According to such training, the learner must find a "rule" for identifying new e-mail.

Conversely, to identify anomalies, each learner receives a large number of unlabelled emails as training. The task is to identify anomalous emails.

Supervised Learning describes scenarios where it is possible to learn from experience by providing more information.

Unsupervised learning does not distinguish between training and test data.
The main feature is clustering, in the sense, collecting unlabelled data with the same specificity.
An example of Unsupervised learning is the k-means clustering algorithm.

2.2.1.a K-Means Clustering

K-Means algorithm is an iterative algorithm that tries to divided the dataset into K pre-defined distinct non-overlapping subgroups (naming clusters) where each data point pertain to only one group.

(Shalev-Shwartz and Ben-David, 2013) explains that finding the optimal $k-means$ solution is often computationally infeasible.
Indeed, in many cases, it is used a simple algorithm, hence the $k-means\ clustering$ relates to the outcome of this algorithm as oppose to the clustering that minimizes the $k-means$ objective cost.

The model of $k-means$ algorithm is structured in this way, by (Athey and Imbens, 2019):

$$C_i = \arg \min_{c \in \{1,\dots,K\}} ||\boldsymbol{X_i} - b_c||^2 \qquad (10)$$

The authors explicate that the feature space is divided into K, cluster.
Then they set K of centroids, $b_1, \dots b_k$ and assign each unit to the cluster that minimizes the distance between the unit and the centroid of the cluster.
As explained before, the number K is difficult to find because there is no direct cross-validation methods in order to enhancement the performance of values.

For all these reasons, it should be better to use an alternative unsupervised methods.

2.2.2 Parametric tests and nonparametric tests

Parametric and nonparametric tests are two classification of statistical procedure
As (Kaur and Kumar, 2015) outlined, the parametric tests relied on assumption on distribution, specifically, parametric data has an underlying normal distribution, the so called Gaussian distribution.
For this reason, authors explains that parametric tests are more robust and demanded less data with respect to the nonparametric tests.

In order to implement the parametric tests, three criterion have to be met:

1. Necessity of normally distribution of the data
2. Same variance and standard deviation among data
3. The data must be continuous

The advantage of parametric tests are described in (Grech and Calleja, 2018), namely that since the distribution is recognized, it is possible to extract inferences about values that lie inside any part of the distribution curve.

Some examples of parametric tests are:

- Student t-Test
- The z-test
- Chi-square
- ANOVA

Furthermore, with regard to nonparametric tests, (Kaur and Kumar, 2015) precises that are methods that do not necessitate a distribution to fulfil the required assumptions to be analyzed (especially if the data are not normally distributed).
Therefore, sometimes it refers to as distribution-free methods.

In a nutshell, nonparametric tests refers to a model function in which there is no dependency on a parameter.

As a conclusion, If the mean more accurately constitute the center of the distribution of the data (which means the distribution is normal), and sample size is large enough, it is recommended to use parametric test.

If the median more accurately describe the center of the distribution of the data, it is recommended to use nonparametric test even if you have a large sample size.

It could also be considered the case in which the distribution is not normal, meaning it is skrewed, hence the recommendation could be to transform data.

It is possible to ascertained the patterns of the distribution in order to understand if the skrewed is due to, for example, outliers.
In such circumstances, it could be better to examine the case in which outliers are cancelled.

Alternatively, when outliers are faraway from the mean, the recommendation could be to analyse if there are errors on the data collection.

Furthermore, another opportunity is to perform the analysis with and without outliers with the aim of discovered if differences display.

By focusing attention again on the nonparametric tests, (Kaur and Kumar, 2015) indicates some examples of tests:

- Sign Test
- Wilcoxon Sign-Rank Test
- Mann-Whitney Test

2.2.3 Function Classes, parametrization and Regularizers

Following (Hastie, Trevor; Tibshirani, Robert; Friedman, 2013, para. 2.7.1):

$$RSS(f) = \sum_{i=1}^{N} \big(y_i - f(x_i)\big)^2 \tag{11}$$

In order to compute the solution of the equation it is necessary to compute the minimization of the function.
By doing so, this means, finding values that verify the equality between two terms.
However, in this case, there is not a unique solution, but an infinitely solution.

The authors mentions that any specific solution could be a bad predictor at test points different from the training points.
In order to obtain a finite solution, it is requirable to define constraints that allows to restrict qualified solution to a smaller set.
The nature of the restriction is decided outside of the data.
These constraints, or restrictions, are codified in the parametric representation of the function.

As (Hastie, Trevor; Tibshirani, Robert; Friedman, 2013) writes, any restriction imposed on f, even if conducted on a unique solution, leaves uncertainty unchanged.
The ambiguity can also be transferred to the selection of the restriction, where infinity restrictions leads to a unique solution.
The strengthened the restriction, the more sensitive the solution is to the particular selection of constraint.

Parametric predictors results in a linear regression function class.
For linear regression the most popular versions are Lasso regression, Ridge regression and Elastic net.

As previously described, the standard linear model (or OLS method) produces poorly outcomes, when there are large multivariate data-set, enclosing a number of variables superior to the number of samples.

In order to overcome this, a solution is the so called penalized regression, where the parameter λ is the penalty.
The penalization creates a linear regression model in which the cost (or penalty) expresses the weight of the high numerosity of the variables.
This is done through a constraint added to the original equation.
This methods, in addition to being called regularization, can also be called shrinkage methods.

The impact of imposing the λ is to shrink the values of the coefficients close to zero.
In this way, variables that add less benefit to the regression, have a coefficient which is close to zero or zero.
λ describes the magnitude of the shrinkage.

2.2.4 The Lasso Regression

(Tibshirani, 1996) proposed a technique called Lasso.
Lasso stands for Least Absolute Shrinkage and Selection Operator.
The aim of this regressor is, as previously specified generally, to shrinks coefficients so that they are close to zero or zero.

The Lasso estimates is defined by:

$$\arg\ min\ \{\ \sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2\ \}\ s.t.\ \sum_j |\beta_j| \leq t \quad [6] \tag{12}$$

Where t is the tuning parameter and it is set to be positive.

This is a linear equation with constraint, where t controls the amount of shrinkage: the larger the value of t (or the larger the value of λ imposing the size of constraint on the coefficients) the greater the amount of shrinkage.

[6] From Regression Shrinkage and Selection via the Lasso, Robert Tibshirani, 1996

(Hastie, Trevor; Tibshirani, Robert; Friedman, 2013) proposed the subsequent passage:

$$\underset{\beta}{\operatorname{argmin}} \{\frac{1}{2} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \}$$

(13)

The constant β_0 is re-parametrized by standardizing the predictors.
In this case the solution for β_0 is $\bar{y}$ and afterward the model is built-in without an intercept.

As mentioned above, it can be interesting to analyse how the entire path of solution changes as λ changed.

An interesting observation made by (Mullainathan and Spiess, 2017), adds further insight to this analysis.
Running this linear regression subject to a constraint on the number of non-zero coefficient, intending penalizing the coefficients, it also process sparse solutions.

From (Zhang and Huang, 2008) it is possible to retrieved a definition for the sparseness concept.
"A model is sparse if most coefficients are small, in the sense that the sum of their absolute values is below a certain level."

Under this statement, there should not be sensitivity to the mutation of coefficients towards zero.

Therefore, for the above, (Zhang and Huang, 2008) proposed to evaluate the model with the sparsity equal to the dimension of the selected model.

The goal is to select a model which approximate the truth well, but measures the sparsity in a proper manner in order to have a suitable measure of performance.

This could be a solution of the problem, but (Mullainathan and Spiess, 2017) continued to analyse the problem by trying to find a cause.

To illustrate the problem, the authors made an example: predicting house prices.
They considered "10,000 randomly selected owner-occupied units from the 2011 metropolitan sample of the American Housing Survey.
In addition to the values of each unit, they also included 150 variables that contain information about the unit and its location, such as the number of rooms, the base area, and the census region within the United States. To compare different prediction techniques, they evaluated how well each approach predicts (log) unit value on a separate hold- out set of 41,808 units from the same sample".

With a view to this data, they selected "ten partitions of approximately 5,000 units each".
This partition allows to observe how variables that are used vary among partition.
The result was that the pattern was not stable much over all the partition.

Authors elucidates that, this was not due to the R^2, because it remains stable over the different partition.
The problem occurred because variables are correlated with each other.

2.2.5 The Ridge Regression

Another type of regression is called Ridge regression which is a popular parameter estimation method.
The popularity of this method, as (McDonald, 2009) writes, is imputable to the fact that it is used to address problems as collinearity, arising in multiple linear regression.

In the case of OLS estimator, the approach is well know: for example, it could be resolved deleting independent variables in order to enhance the matrix of correlation.
In the case of Ridge regression, the requisite of removing independent variables, does not exists.

(Hastie, Trevor; Tibshirani, Robert; Friedman, 2013) proposed the same problem, as for Lasso:

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \quad s.t. \quad \sum_{j=1}^{p} \beta_j^2 \leq t$$

(14)

Where t is the tuning parameter and it is set to be positive.

As for Lasso, t controls the amount of shrinkage and is a one-to-one correspondence with λ.

By imposing the constraint:

$$\underset{\beta}{\operatorname{argmin}} \{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \}$$

(15)

In this case, it is observable how Lasso problem constraint is set very similar to the Ridge problem constraint.

In Ridge regression, there exists the difference between the realized y and the predicted y subject to $l2$ norm ($l1$ norm was the difference inside the Lasso estimation method) and the penalty parameter, λ, is interpretable as cost on the sum of squares.

Unlike the Lasso, which produces sparse solution, in this case coefficients are likely to be distributed.
Rather than picking out a single coefficient that is non-zero, this solution tends to produce coefficients that are more equally distributed, meaning that in doubt it would produce coefficients on all the variables that are non-zero rather than producing sparse solution.

This could be interpreted as observation posterior as bridge regression, in the sense that represents what would be the outcome if a regression on linear model is done with normal data comparing with the computation of Bayesian posterior, getting exactly the same solution.

As (Athey and Imbens, 2019) emphasized, the difference with formal Bayesian approach, is the coefficient λ that, in the case of Machine Learning algorithm, is chosen through out-of-sample cross-validation, while in Bayesian learning is chosen through prior distribution.

2.2.6 The Elastic Net Regression

This regression is a combination of Lasso and Ridge regression.
If the Lasso decides, indifferently, the choice between a set of correlated variables; the Ridge tends to reduce the coefficient of variables correlated each other's.

The form of Elastic net regression is defined in the following (Hastie, Trevor; Tibshirani, Robert; Friedman, 2013):

$$\sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)\beta_j^2 \right)$$

(16)

It is observable how the structure of the model differs for Lasso regression and Ridge regression.
The first variable induces a sparse solution of mean feature coefficients and the second variable induces averaging of strongly correlated features.
Having an optimization algorithm that solves the structure, the feature class, the regularizer, and the feature class under regularization constraints are not only unique to linear regression, but these principles are used by many supervised learners.

2.2.7 The Regression Trees

A regression Tree is a nonparametric predictor and it is defined as a decision tree that is used to predict continuous valued outputs instead of discrete outputs.
The process behind the algorithm is recursively partitioning the data space by selecting certain points that best splits the data-set. The algorithm doing this in a repetitive forms, shows up in a tree-like structure.

An example is the one represented by (Loh, 2011):

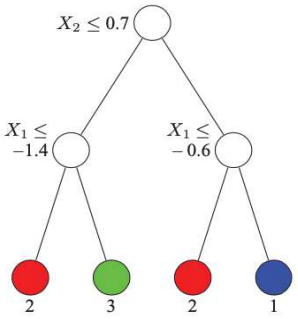

Figure 1: "Decision tree structure for a classification tree model with three classes labelled 1, 2, and 3. At each intermediate node, a case goes to the left child node if and only if the condition is satisfied". Source: (Loh, 2011)

By deciding whether to go left or right based on the value of the data, it is possible to represent much more complex relationship that includes interaction between variables in way that would be very hard to model using linear regression.

In the case of regression trees, the problem is the same as before: if instead of a very complex linear regression, which can cause overfitting, it is possible to limit the complexity of these regression trees, for example, by limiting the number of trees levels to allow or limit the number of individuals belonging to one leaf of the regression tree.

The advantage of considering a model like Regression trees, is that if in one hand it is possible to consider OLS regressor as a function globally constraining (meaning the effect of one of the variables is the same, no matter the values of the other variables), on the other hand considering a regression trees it is logically possible that can model interactions much better and it can cut the variable space of features into a places that is not globally constrained but instead produces a very local solutions.
Regression Trees is modified from a focus on estimation of regression function to classification tasks.

Classification means valuating the efficiency of a model by looking at goodness of fit in a held-out test set.

The difference between the regression case and the classification case is called impurity function.

A more common impurity function is the Gini impurity.
From (Suthaharan, 2016)

$$-\sum_{i} p_i(1-p_i) \tag{17}$$

Where p_i is the probability of the i^{th} event.

The abbreviation of Gini stands for generalized inequality index.

(Athey and Imbens, 2019) explain that this impurity function is minimized only if all units in a leaf have the same label, while it is maximized if the shares are equal to $\frac{1}{M}$ (where M is the number of shares).
As described for the others regression model, the regularization occur applying λ, the penalty term on the number of leaf in the regression.

It is important to pointed out, as before, that the structure of supervised learners is very general. It usually contains some functional class, regularizer and optimization algorithm but also includes for example neural networks.

Regularization may for example work by constraining the number of layers that neural network has, or by constraining the connects that the neural network has or restricting the amount of optimization that goes into fitting it.

2.2.8 The Neural Network Regression

Neural network, compared with previously model, is a mixed predictors.
Neural network, as (Athey and Imbens, 2019) describes, is another general and flexible approach to estimating regression function. The problem of this method is that it require a significant amount of model selection to managed, in a proper manner, with respect to other methods.
Neural network is a two-stage regression represented by a network diagram.

2.3 Enhancement performance techniques

This paragraph describes techniques that allows to improve the performance of Machine Learning algorithm outlined before.
As specified, Machine Learning methods use data-driven model selection.

The problems arises from Machine Learning are primary classification problems with the intention to categorized objects inside a unique group.

For instance as (Schapire & Freund, 2013) highlights, the spam-filtering is a classification problem in which specified e-mails are catalogues as “spam or ham”, in this way the algorithm segregates and labels the entire sample.
The outcome of this algorithm is a predictor or also called from the authors, “classifier or hypothesis”.

Basically, the researcher offers a list of features but since the form of the function is determined as minimum function of the data, the process is better specified as an algorithm that estimates different model and then select the one that maximize a benchmark.

An error rate measures how often a classifier makes incorrect classifications, which is the quality of a classifier.
It is necessary to develop a test set, a collection of test examples in order to accomplish this.
Classifier predictions are compared with correct classifications for each of the test examples in order to evaluate its performance.

(Schapire & Freund, 2013) defined test error the incorrect classification of the classifier, and training error the incorrect classification of the training set.
They called, instead, accuracy, the correct predictions.

Generally, as (Schapire & Freund, 2013) emphasizes, the training and test sets comes from the same random source when designing and studying learning algorithms.

Specifically, the authors assumes that the training and test sets are randomly drawn from some fixed but unknown distribution over the space of labelled and, furthermore, that both the training and test are generated by the same distribution.
Classifier generalization error measures the likelihood that a random test sets is generated by the unknown distribution.
The aim, then, is also to create low generalized errors, with the purpose of having a predictor that is the most accurate as possible.

Very explicit in this respect is (Athey, 2018).
There is a compromise between the number of covariate included in the model, hence how much variables are correlated each other's, and the overfitting condition, where the number of variables are greater than the number of samples.

In order to analyse the goodness of fit the model, there are different techniques with the aim to overcome this trade-off between the expressiveness and the overfitting.

As (Athey and Imbens, 2019) clarify, the performance of these methods can be distinguished between:

- Sample splitting
- Cross-fitting / Cross-validation
- Out-of-bag prediction
- Leave-one-out estimation
- Boosting
- Orthogonalization

Or as (Athey, 2018) illustrates:

- Stochastic Gradient Descent (SGD)

Based on these distinctions, the one that will be insighted are Stochastic Gradient Descent (SGD), Boosting, Orthogonalization, Cross fitting / Cross validation.

2.3.1 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (or SGD) is used in many types of model, like the neural networks and it operates in this way:
Firstly is a methods that optimized the objective function, with respect to the parameters.
If each parameter corresponds to a single observation, Stochastic Gradient Descent assesses observation as an average of the gradient.
In this way the estimation would be unbiased.

More specifically, (Shalev-Shwartz and Ben-David, 2013) clarifies that since the loss function is unknown and the distribution is unknown, Stochastic Gradient Descent generates the optimization using a random direction, specifying also "as long as the expected value of the direction is the negative of the gradient".

Negative gradient meaning find the lowest value of x such that y is minimum; where y is the objective function in which the gradient descent algorithm acts.

Stochastic Gradient Descent is an efficient techniques due to its simplicity and for the fact that it can be applied when any other model based on empirical risk cannot be applied.

The steps of the algorithm are very complicated but they can be summarized into six steps:

1. Compute the gradient of the objective function, meaning the sum of partial derivatives with respect to each individual features (or parameter)
2. Take a random value for the features
3. Bring up to date the gradient function by entering the values of the features
4. Computing the step sizes for each features which means multiply the gradient for the learning rate. This rate influence the convergence of the algorithm, the higher is the learning rate the higher the possibility that the algorithm makes a great jump and miss the minimum point.
   It is reasonably, for what has been said, to bring a lower learning rate.
5. The new features (or parameters) = old features – the size of the step
6. Reiterating these steps many times until the gradient is equal to zero.

The main problem of this technique is that it generates a model which is unbiased but the gradient will be noisy.

Even if the existence of this problem, Stochastic Gradient Descent enables to improve the performance of very complex estimation that would be unmanageable using traditional estimators.

2.3.2 Boosting

Boosting is another techniques that enable to enhance the performance of simple supervised learning methods.

In order to understand how this method works, it is possible to use the example of the spam or junk e-mail.
Following the logic structured by (Schapire & Freund, 2013), the intent is to build a spam filter. In this way only the non-spam e-mails are filtered from the spam e-mail and categorized automatically into two classes: spam or ham.

This suggest that, Machine Learning algorithm should recognize, from the recursively categorizations, a prediction "law" that classified in a properly manner, the two classes, having as a result an accurate predictor.

In particular, the authors brings about an example: "if it contains the word Viagra, then it is probably spam", but suggests also that "A rule that classifies all email containing Viagra as spam, and all other email as ham, will very often be wrong".

For this reason the algorithm produced a weak rule and it could be interpreted as a "weak learning algorithm".
But it would be possible to extract a set of rules from the dataset by repeating the algorithm on a different subsets.
Boosting consists of somehow combining there weak rule of thumbs into one group whose predictions will be fairly accurate overall.

The weak learning program faces two critical problems: the first one is that it is necessary to choose the best useful rule of thumb from the collections of e-mail.
The second one is that it is necessary to develop an accurate predictor from the stored rules.

Boosting algorithm does exactly that: formulate an effective process in which an accurate rule by combining "weak learning".

In order to better understand the model, (Athey and Imbens, 2019) proposed a simplified explanation.
The model referred is a regression tree with three leaves, similar to the Figure 1, based on two splits.
Boosting is able to improve the performance of the regression along the following line.

The problem can be set up as previously described, namely finding the prediction of the variable y from the variable x, in a way that the average loss between the predicted y and the true y is as low as possible.

In this case, the author defines this passage as:

$$Y_i - \widehat{Y}_i^1$$

(18)

Which is applied to the tree model.
Now, using the equivalent regression tree (also called base learner), the same process is applied to the two-split model, having a new outcome (or residuals) and in this case:

$$Y_i - \widehat{Y}_i^2$$

(19)

Doing this process iteratively, the outcome is a predictor that update and re-estimate every time the residuals.

An advantage of Boosting is that it can be used for other regression models.

Generalizing the Boosting method, taking into consideration the regression tree model with N splits, then it prove to be that the sum of functions of N of the initial features is almost accurate to the predictor.
Therefore, $N = 1$ it is possible to estimate any function that is supplemented in the features.
$N = 2$, stands for estimate any function that is supplemented to the original one.

A more detailed explanation is given by (Hastie, Trevor; Tibshirani, Robert; Friedman, 2013).
The loss function is defined as:

$$L(f) = \sum_{i=1}^{N} L\big(y_i, f(x_i)\big)$$

(20)

Same as before, the logic is to minimized the loss function with respect to the function itself.

The numerical optimization is:

$$\underset{f}{\operatorname{argmin}}\, L(f)$$

(21)

Where,

$$f = \{f(x_1), f(x_2), \dots, f(x_N)\}^T \quad (22)$$

Consequently f refers to the value associated to the function $f(x_i)$.

The authors continue the analysis by pointing to different types of gradient, among them:

- Steepest Descent, where g_m is the gradient of the loss function and it has a negative sign which means the direction is negative along the function (that is the lower value of the objective function) but it is much steeper than the f
- Gradient Boosting, where the value associated to the function f are not independent but constrained

2.3.3 Bootstrap and Bagging

Bootstrapping is another methods that allows to improve the performance of an estimation. (Schierle, Martin; Schulz, 2007) asseverate that "Bootstrapping algorithms can be seen as iterative supervised classifiers, which are initially trained on a very small training set consisting of several examples, which is called seed S_{Seed}. After application of the classifier and an additional evaluation and filtering step of the results S_{New} which leads to S'_{New}, the classifier is trained on the merged set of S_{Seed} and S'_{New}".

The process of the Bootstrapping can be summarized in different steps:

1. Defining the training set S_{Seed}
2. Cluster the data and then extract only a features (or parameters)
3. Generalized this derivation with a pattern and apply this trend to the cluster of the data
4. Weight the S_{New} using information and then filtered using a given constraints
5. Then $S_{Seed} = S_{Seed} \cup S'_{New}$
6. Iterate all these steps again

Bagging instead, as (Varian, 2014) states, "involves averaging across models estimated with several different bootstrap samples in order to improve the performance of an estimator".

Bagging was introduced by (Breiman, 1996), the idea as (Zeng, Chao and Wong, 2010) summarize is that it wants to equal produce different classifiers and merge them on arbitrary redivision training datasets.

According to (Breiman, 1996), N is the dimension of the datasets, therefore there is a causal drawing in which every dataset is created.

Also Bagging, as Boosting are mainly used in regression trees.

2.3.4 Bumping

The authors of (Hastie, Trevor; Tibshirani, Robert; Friedman, 2013) describes, in a timely manner, Bumping method.
Bumping, as the other methods, allows to find a better performance of the model, avoiding from imprisoned into a bad solutions.
More specifically, Bumping method deriving from Bootstrap method, thus the squared error is obtained from :

$$\hat{b} = \underset{b}{\operatorname{argmin}} \sum_{i=1}^{N} [y_i - \widehat{f^{*b}}\,(x_i)]^2 \qquad (23)$$

Where $\widehat{f^{*b}}$ is original objective function deriving from the bootstrap procedure.
What this methods does is to fitting the process into an acceptable model area.

2.3.5 Orthogonalization

Orthogonalization is another type of method that works very well in Machine Learning with the aim of creating an accurate predictor.

Generalizing, orthogonality is defined as variables that are uncorrelated each other's but also uncorrelated with the error terms.
In the sense, that in the model, variables are all independent.
Therefore if one of them are correlated, the model is non-orthogonal.

The correlation is define as multicollinearity among the data.

(Shankhar, 2020) states that orthogonalization allows to reduce the time for the testing part and does not create propagation of changes in the system.

The author provide an example, using an old television with a lot of knobs that can be tune to adjust the picture in various ways.
Thus for these old TV sets, maybe there was one knob to adjust the vertical height of the image, or another knob to adjust the width, the orders or the rotation.
Designers have to make sure that each of the knobs had their residence of the interpretable function.
Orthogonalization refers to the TV designers had designed the knobs so that each knob has to do one thing.

Another example concerns learning to drive a car. A car has three main controls which are steering.
Steering wheel decides how much go left or right acceleration but also breaking.
It makes it relatively interpretable that the different action corresponds to different controls.

It is possible to reflect in ahother way: if someone were to build a car so that there was a joystick where one axis of the joystick control 0.3 times the steering angle minus 0.8 times the speed, and there is a different control that discipline two times the steering angle plus 0.9.

In theory, by tuning these two knobs it is possible to get the car to steer the angle and the speed wanted. But it's much harder than if having just one single control for governing steering angle and separate distinct set of controls for controlling your speed.

The concept of organisation refers to this example.
Thinking in one dimension controlling the steering angle, and another dimension as the speed, then want one knob.
However having to control a mixes the two together, as both steering angle and annual speed, then it becomes much harder to set the car to the speed and angle wanted.

Orthogonalization can be interpreted as aligned with the things that want to control.

In order to compute all the passages in a great manner, it is necessary that four assumptions are true.
First is that to make sure that the fit is doing well on the training set. Thus, the performance on the trading set needs to pass some acceptability affection for some applications. This might mean doing comparably to human level performance.
Then it is necessary to do well on the dev set.
Afterward, it is necessary to do well on the test set and finally do well on the test set cost function results in the system performing in the real world.

2.3.6 Cross-validation

Following (Dreiseitl and Ohno-Machado, 2002) there are two possible benchmark that indicates how a model can be categorized:

- Discrimination
- Calibration

The former checks how data are split while the latter quantify the estimation taking into consideration the probability of the evaluation.
In particular the probability can be computed from a dataset that is not used for the estimation, in this way the predictor would be unbiased.

Cross-validation as (Yadav and Shukla, 2016) indicates "is the most commonly used method for predictive performance evaluation of a model, given beforehand or when it is developed by a modelling procedure".
All the information is split in two parts, the training set and the testing sets.
The model is trained taking off some example out of to test.
For the overfitting problem, the performance on the training set is not a good predictor for the data that are not disclosed.

As (Bharadwaj, Prakash and Kanagachidambaresan, 2021) states if the data are substantial, in order to have a good predictor in terms of performance, it is necessary to compare the values of the training data with independent data, called "validation set". Afterwards the performance can be evaluated reserving a "test set".

The problem arises because many times the availability of the data is limited.
To overcome this problem, the authors defined a “cross-validation” solution.

The critical point is that it is necessary to divide the model in different partition choosing a “validation size”.
Depending on the type of validation size there could be the possibility of having over-fitting or under-fitting problem.
For this reason the selection of this variable is determinant.

This conundrum can be explained more clearly using an illustration :

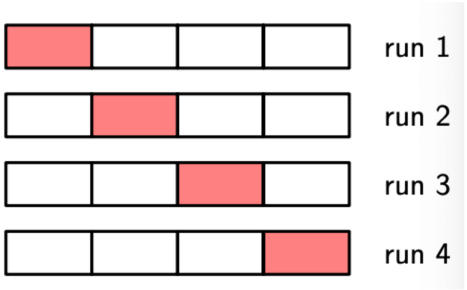

Figure 2: S – fold cross validation, Source: (Bharadwaj, Prakash and Kanagachidambaresan, 2021, p. 33)

The dates are proportioned into $\frac{S-1}{S}$ for the training set using all the availability information in order to compute a good predictor to measure the performance.

In Figure 2, the number of partition is four (S), the residual data set (S-1) is used to training the models.
Repeating this procedures for all possible choices for the red blocks (hold-out-set), then the outcome is finally averaged.

There are different problems inside this technique:

1. The case in which the computation is really expensive because the training arise by a factor of S
2. The case in which, for one model, there are several parameters that allows to evaluate the performance. This creates also a potential overfitting issue, in which the number of variables are greater than the number of sample.

Usually in order to correct these problems, as explained previously, the possibility is to add a penalty (λ parameter) that offset for the overfitting.

Due to (Yadav and Shukla, 2016) the main goals of the cross-validation technique are:

1. Using the algorithm to foresee the performance
2. Finding the best algorithm, matching different type of data

Cross-validation can be called also K-fold Cross-validation, where data are divided according to a k values.
Repeating this k-times modifying the test part until every sample point at some point in order to do evaluation on.
In this case the model is called repeated k-fold cross-validation.

As for all the other models described, the final purpose is to find an accurate predictor of the model.
In this case, the higher is the number of estimates the higher will be the accuracy of the model predicted.

Cross-validation can also be seen from another point of view, suggested by (Newey and Robins, 2018).
This study uses cross fitting combined with "fast remainder rates".

These indicators are distinctive in that because they converge towards zero.
Their size is decreased by cross-fitting because it deleting the bias observations.

According to the authors, it is possible to illustrate how cross-fitting allow to reach fast remainder rates.

Considering the expected conditional covariance:

$$\beta_0 = E[Cov(\alpha_i, y_i \mid x_i)] = E[\alpha_i\{y_i - \gamma_0(x_i)\}] \quad (24)$$

Where $\gamma_0(x_i) = E[y_i|x_i]$

The estimator that results has this form:

$$\bar{\beta} = \frac{1}{n}\sum_{i=1}^{n} \alpha_i\{y_i - \hat{\gamma}(x_i)\}$$

(25)

Where $\hat{\gamma}(x_i)$, as (Newey and Robins, 2018) underly, is affect by an "own observation" which means that this parameter is biased.
The correction is done by substitutes the variables $\hat{\gamma}(x_i)$ with $\hat{\gamma}_{-i}(x_i)$.

The estimator in this case becomes:

$$\hat{\beta} = \frac{1}{n}\sum_{i=1}^{n} \alpha_i\{y_i - \hat{\gamma}_{-i}(x_i)\}$$

(26)

This correction is exactly the cross-fitting methods which allows to improve the performance of the estimators, deleting the bias.

In conclusion, with prediction you can check and hold outside whether the predictor got it right. With estimators you do not have β grounds truths.
Prediction is empirically grounded because when someone gives you a predictor you can check whether it is working, while estimation is not verifiable.
If you trust the assumptions there is no way to check them.

With prediction it is possible to see the actions.
This luxury is not possible to have when researchers compute an empirical studies that relies on parameter estimation, where we need other methods to convince ourselves that we find is reasonable but it is not possible to observe directly in the data.

By making a recap, Machine Learning often involves three steps:

1. Choosing a flexible function class
2. Limit regularization
3. Understand how much to tuning the parameters

There are some researcher choices left that are very central to any good empirical application including loss function do optimize for.

How is it possible to manage the data making sure that you keep the right out of sample experiment so you can run at the end a valid tests.
Choosing which features to use in data, so how to pre-transform your data how to deal for example missingness, and which function does not regulate to choose.
Machine learning is a powerful framework to find good predictors among a class of function chosen.

3 Machine Learning and Traditional Methods

There is a lot that has already been known in statistics and econometrics, that is, new names for similar things.
Specifically the idea that we can do better at predicting in high dimension than using some simple unbiased estimator like OLS.
Most famously associated with a paper by (Bickel *et al.*, 1998) that showed that as soon as liner regression has at least three covariates, it is possible to do strictly better than predicting if there is no necessity about unbiasedness.
Random forest themselves, come out of statistics associated with the (Breiman, 2001) and there is a large area in econometrics that studies non and semi-parametric estimation and Bayesian estimation.

There are news about this world that goes beyond these innovations in statistics and econometrics.

Some of the ingredients that have to come together are:

- Availability of data that involves very complex data that involves the ability to access that data in a structured way, for example for text data, for image data, but also just for very complex social science data.
- Ability to compute those things and fit in network for example or to even work with very large dataset on other server or in local machine
- Innovation and function forms that we use. Neural network have been around for long time, for example, but only with this combination of data computation and specific function form, that actually work for specific tasks, they have become as successful as they are these days

It is plausible to use that prediction function in applied work but this prediction may not be necessary for the goal.

Machine Learning could be interpreted in a naïve way.

Normally, the interested part is to compute inferences on the relationship of the y variables to the x variables, understanding how does y changes when changed a specific x variable.

This may include causal inference questions, where holding everything else constant, the x changed.
That of course is tempting when the output has a common form (like Lasso).
Lasso does not conveniently give a linear regression, but gives also linear regression coefficient where zero can takes many places.
It is tempting understanding which variables could be chosen and how the outcome is affected in some way.

It is important to understand the degree on which this come with, using a simple parameter estimation.
To which degree does this come with guarantees of unbiasedness of consistency, meaning that the estimated coefficients in large sample approximates the true coefficients.
In other words, to which degrees inferences of robust to violations of the assumptions about the relationship of y and x that may embedded in the model.

As a remainder, Lasso puts penalty on the coefficients in a way that does not only shrink the coefficients but it also makes many of those coefficients exactly zero.
In that way its capitalist gives much of the predictive power, it associates with a single variables, thus, it does not spread out predictive power between coefficients but instead selects a few that are non-zero.

An example could be reasonable in order to understand what concretely mean.

An example of OLS regression,

$$default = \alpha + \beta_1 income + \beta_2 age + \beta_3 education + \beta_4 creditscore + \beta_5 x_5 + \cdots + \beta_{27} x_{27} + \cdots \beta_{80} x_{80} \quad (27)$$

comes from studying fairness considerations in using Machine Learning for underwriting.

Basically banks used Machine Learning in order to understand at who should get credit. Banks may resort to those more complex tools because they are facing a simple prediction tasks of predicting, based on credit file, whether somebody is likely to default.

Therefore, instead of running OLS of many variables, Lasso runs OLS but it puts a constraint on it, meaning that many those variables will be zero.

One way in which it could be possible to test is whether the information is a good prediction about the relationship of default of those variables.

Empirically it is possible to test doing this many times on similar data.

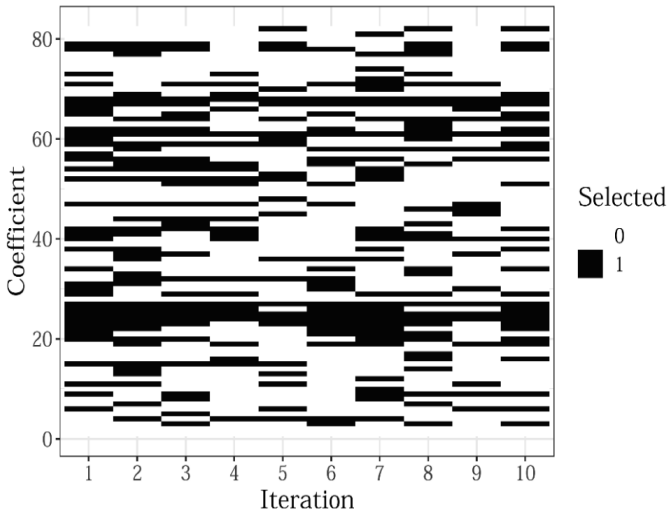

Figure 3: included predictors in a lasso regression across ten samples from the same population, Source: (Gillis and L, 2019)

The focus is understanding the implication of those algorithm.

Apart this, (Gillis and L, 2019, fig. 2) describes an econometric example, selected in Figure 3. On the x-axis, there are the iterations in taking sub sample of the same data, on y-axis the coefficients on in total 80 variables.

It is clear that, while the Lasso that is fit on data from the same balance growth path multiple times, it has some persistent structure where some of those variables are chosen very time. Many of them look more noisy and there is no strong structure in many other coefficients.

That means that is consistent, meaning converge to the true coefficient.
The algorithm now learned from the lasso which variables specifically matter.

Why in this case the selection change so much from one draw to the other?

$$default = \alpha + \beta_1 income + \beta_2 age + \beta_3 education + \beta_4 creditscore + \beta_5 x_5 + \cdots + \beta_{27} x_{27} + \cdots \beta_{80} x_{80} \quad (28)$$

The fact is that could be that variables are correlated:

$$default = \alpha + \beta_1 income ~~+ \beta_2 age~~ + \beta_3 education + \beta_4 creditscore + \beta_5 x_5 + \cdots + \beta_{27} x_{27} + \cdots \beta_{80} x_{80}$$

$$age \approx f(income, creditscore, \dots, x_{27}, \dots) \quad (29)$$

In extreme case, like the case in which it is possible to learn about somebody's age from a combination of other variables.
Different ages, different financial history, thus this is a higher level of learning from that variables.
This means that it is possible to write a function that does not include age but still contains the same predictive information and from a prediction point of view this does not really matter.
Variables gives the same information, assuming that the relation are approximated well by linear regression.

Indeed, if all that concerns, is good prediction this is not something that could create an alarmed. In the opposite case, where it is necessary to care about estimation or to understand whether certain people are treated differently, then it becomes a primarily important to distinguish.

A numerical example about the Lasso biases from (Blumenstoc, 2021):

$$y = 2x_2 - x_3 + \in \tag{30}$$

Where Lasso is fitted with x_1, x_2, x_3 on the function 30.

Assuming that two variables are highly correlated, in this case x_2 and x_3 , both included in function.
It could be that, one of those coefficients is simply set to zero because it is more efficient for the Lasso to only use one of the coefficient to express the relationship on both.
Thus, this means, $\widehat{\beta_3} = 0$.
Thereby, it would be the case if the price that Lasso puts on complexity is relatively high, hence adding a higher coefficient would be inefficient and the correlation of x_2 and x_3 is relatively large.

Replacing this function by $y = x_2$, it approximately gives a good enough prediction at a far lower cost in term of the size of coefficient.
It can also be that Lasso includes some variables that are not even in the original function.
Assuming, for example that there is a variables x_1 that is not in the true relationship between x and y but it is very correlated with y in a way that is not structurally correct, meaning there is correlation, but it is not the true functional form the Lasso may refer to just use a single variable x_1 to include both x_2 and x_3 in order to capture the relationship, $\widehat{\beta_1} \neq 0$.

Finally even if a variable, that is truly in there, is included, it does not mean that they are not also biases.

Hypothesize the case in which there is an omitted x_3, but the variable x_2 is included in the model.

In that case the coefficient getting on x_2 will be heavily biased because there are omitted correlated variables.
The Lasso, rather than just being prone to biases that OLS, would also be prone to actively creates omitted variables bias, because it actively excludes some variables there may be some other variables that therefore have a bias coefficient.
Even If x_1 and x_2 are both selected, meaning even if there is exactly the right variables selected, then the Lasso would still bias them towards zero just because it puts a cost on the coefficients.

The main take away is that Lasso can be biased in ways that includes variables that are not in the true relationship, that excludes variables that are in a true relationship, and even if it does include the right variable, it may still be that it suffers from additional omitted variable bias. Lasso decides to extrude other correlated variables which are exactly those that lead to omitted variable bias.
Hence, in high dimensions at least empirically, these correlations will be ubiquitous and for this reason Figure 3 seems like a barcode where the chosen variables switch around very widely.

Machine Learning can also be thought as a prediction tool compared to a simple estimation tool like linear regression.
The main advantage is that Machine Learning is for prediction, caring about out of sample loss minimization, while the linear regression is typically built to do an estimation making inference on coefficients.

In high dimension it is possible to have a very good precisions, even when coefficients are unstable or biased or inconsistent, because the main concern is about prediction quality and in many cases that is not necessarily a problem for prediction quality.

Specifically many functions that look very different can have very similar prediction properties and it is very hard to distinguish them.
The features discussed make up a successful prediction algorithm, makes the estimation hard, namely the fact that it is possible to use the data in order to select among a large class of functions making it very hard to determine the estimation properties and getting a valid estimation.

(Mullainathan and Spiess, 2017) report an interesting definition of prediction and estimation.

Prediction is a good fit of $\hat{y}$ to y on data from the same distribution rather than some other distribution with some modification.
Achieving a good prediction really depends on the features observable which means that the distribution stays the same as the distribution already seen.

On the other hand Machine Learning does deliver estimation because the prediction function gets close to the true function, in a sense of, it produces small loss.

It is possible to put the difference between the truth and the prediction function.
More precisely, estimation meaning asking whether the estimator is consistent, thus whether the approximation of the function cares about distinguishing between the individual parts of the prediction function and not just its prediction quality.
Estimation consistency meaning the estimates converges to the truth, not just the overall loss converges.

It is possible to bridge the gap between the Machine Learning and estimation, using an example.

In order to obtain a valid estimation, it is necessary to assume a low dimensional, meaning assuming that data are not complex.
Applying the Lasso to use linear regression problem and recover the true coefficient.

A very classical statistics example from (Knight and Fu, 2000).
The true model is a linear regression model,

$$y = \beta' x + \varepsilon \tag{31}$$

fixing the number of coefficients, k and let n to go infinity.

As long as λ fulfils a certain assumption, that is $\frac{\lambda_n}{n} \rightarrow \lambda_0 \geq 0$, the true coefficient is recovered.

The result could be summarized saying that in low dimensions, this model works.

However, the analysis is concentrated in applying the case in high dimensionality.
This is not well modelled by just holding k fized and let n grows.

Therefore for linear regression in high dimension, it is not just sparsity.

(Zhao and Yu, 2006) is the most important paper in the literature that expresses the assumptions under which the model is consistent.

The assumptions:

- Sparsity,

$$s_n = \left|\left|\beta^n\right|\right|_0 \qquad s_n = \mathcal{O}(n^{c_1})$$

$$c_1 \epsilon [0,1) \quad X_1^n = \left(X_{(1)}^n \ldots . X_{(s_n)}\right) \quad X_2^n = \left(X_{(s_n+1)}^n \ldots . X_{(k_n)}\right)$$

(32)

- Separation,

$$n^{\frac{1-c_2}{2}} \min_{i \in \{1,\ldots,s_n\}} |\beta_i^n| \geq M_3 \qquad c_2 \epsilon \, (c_1 + c_3, 1]$$

(33)

- No collinearity,

$$\inf_{||\alpha||_2 = 1} \alpha' \frac{(X_1^n)' X_1^n}{n} \alpha \geq M_2$$

(34)

- Regularization asymptotic

Sparsity meaning that there is a separation between the variables that are in the model and the variables that are not in the model.
The variables that are in the model have nonzero coefficients, the variables that are outside the model have zero coefficient.
There are few additional assumptions: the coefficient in the model are all relatively high, hence they do not go to zero fast enough and the coefficients outside the model are either very small or they are exactly zero.

Then the third assumption, describes that the x's that can be included in the model can not be too collinear.
This means that the x's that are in the model can not to be too correlated because otherwise it is not possible to distinguish between different values in the model.
Then the last one describes that correlation goes beyond the variables in the model.
Up to now it is reasonable to just apply this model when the estimation is simple, when the coefficients are not too small and when the x variables are not too collinear working also with transformation.

However there is an additional strong assumption, introduced by (Zhao and Yu, 2006) which also controls the covariance between the variables that are included in the model and those variables that are not included in the model.

The irrepresentable condition:

$$\left|\left|((X_1^n)'X_1^n)^{-1}(X_1^n)'X_2^n\right|\right|_{1,\infty} < 1 - \eta \tag{35}$$

The irrepresentable condition states that whenever there is a regression of variables that are not included in the models on the variables that are included in the model then the regression coefficients has to be very small.
In other words, the correlation between the included and the excluded model, has to be very small in order for to be able to correctly assign which part belonging, non-belonging to the model.

Finally, it is interesting mention a negative results, (Leeb and Pötscher, 2008) that states that is also generically hard to do any inference after selection.

Machine Learning provides quality predictions but the prediction quality, while it comes with guarantees from the holdout, it does not typically come with estimation consistency or guarantees about the interpretation of the coefficient of that model.

Hence, by itself, it is not possible to get structural interpretation or counter factual extrapolation like causal inference out of a machine learning model.

As a side note it is also very hard to do inference, as the bootstrap inference because typically do not work in those cases.

This is not just an issue of implementation, meaning that it not just concerned that the model used is the wrong method to do inference in high dimensions but it is inherently limitation.
The consistent estimation is inherently challenging considering many variables and many possible functions.

Prediction quality can be observed, estimation quality can not and therefore will be much harder to do high dimensional inference and coefficients in order to still get good predictors.

There is an emerging econometric playbook that bridge that gap by distinguishing between the prediction part on one side from the estimation part of another.

One example is an IV estimation, where the first stage is as a prediction part, the second part stage is as an estimation part.
Combining these two by using smart sample splitting that guarantees certain estimation qualities and while also leveraging high prediction quality.

Machine Learning and the relationship to standard econometrics not just offering a different answer to the same question but instead they provide answer to a different questions where Machine Learning answers a prediction question, while many tools used in applied econometrics answer to an estimation of $\hat{\beta}$ questions.

Therefore, (Athey, 2018) states that are some exciting opportunities going forward for research.

1. Applied research opportunities where it is possible to use new data, tackling new questions by using improved methods and being more systematic about data driven choices by adapting methods or paradigms for Machine Learning

2. There are also very exiting econometric questions that kind of bridge the gap between $\hat{y}$and $\hat{\beta}$: how can we use good predictors if we have parameter estimation problem by effectively using the Machine Learning part for some nuisance components.

3. Recently is also going beyond the simple supervised learning, so going beyond the simple prediction Machine Learning also offers tools that work very similarly on, for example, clustering or dynamic experimentation.

4. Finally an important opportunities in policy and theories that ask the question on what happens if individual agents and the model interact with the use of Machine Learning, like transparency and fairness of Machine Learning that are inherently think required input from economists because they are inherently about strategic interactions of agents and about welfare consequences and therefore require all of our input.

4 Applications of Machine Learning Methods

4.1 Poverty

To highlight the scale and scope of the covid crisis in low and middle income countries, of course Covid-19 has caused all sort of pain and hardship everywhere including U.S. but what is different in low/middle income countries is that there is a very large populations already living at or below a subsistence level.

The lockdown orders that are design to stop the transmission of the virus also force a lot of people to stop working.

There are a lot already vulnerable households which take away their primary means of subsistence and this has lead a massive rise in food insecurity.

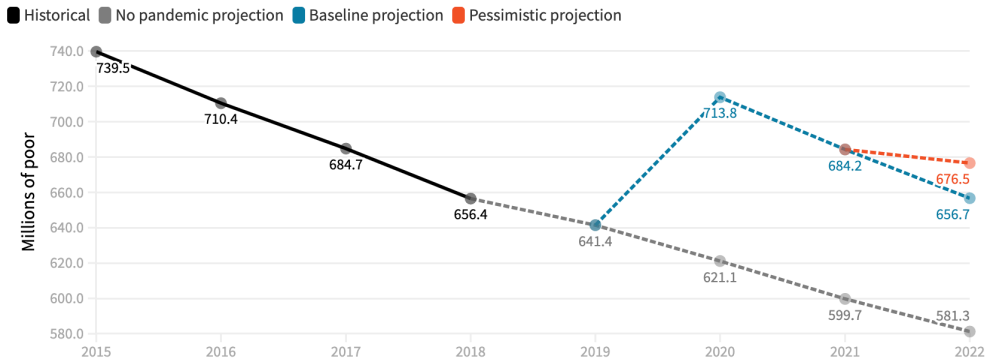

Figure 4: Global poverty is increasing, Source: (Lakner *et al.*, 2022)*, Poverty & Inequality Platform (PIP), Macro and Poverty Outlook.*

For the first time, in a few decades, global extreme poverty is on the rise and estimates of on the order 100 million new poor now in extreme poverty.

There was also a massive humanitarian response being mobilized to combat the consequences of the pandemic.

From (Wb *et al.*, 2020) there were about 1400 new targeted assistance programs in response to Covid-19.

These are targeted programs meaning that are not enough resources for universal basic income, or universal transfer.

There are 100's of millions of families receive some form of targeted assistance.
The cares act provides assistance to families earning less than 180.000 year.

The question is how is the targeting accomplished? In the U.S. the government has the internal revenue service that collects tax records for the vast majority of U.S citizen and thus they are using that as a rubric to determine who has the greatest need.

The challenge, looking at middle income countries, is that huge portions of the population are not reporting taxes, earning regular income.
In general, a lot of these countries lack up-to-date information on living conditions of their population.

From (Jerven, 2013):

Country	***Year of last census***	***Year since last census***
Somalia	1986	28
Congo, Dem, Rep. (planned for 2015)	1984	30
Eritrea	1984	30
Afghanistan (2011 on-going Socio Demographic and Economic Survey by province)	1979	35
Lebanon	1943	71

Table 1: Countries with outdated censuses, Source: (Jerven, 2013)

In the middle of pandemic, trying to minimize face-to-face interaction, it is really impractical to image going out and gathering the data that necessary to use.

How machine learning when applied to non-traditional data can be used to measure and target poverty?

There was a policy mandate to prioritize the poorest regions of the country.
It is not possible have any dataset and data that allowed them to assess the poverty or the socioeconomic status of even the normal geographic regions of the country.

(Yeh *et al.*, 2020) states that only half of the poorest countries have completed a census in the past 10 years.

Researchers thinks about how it is possible to use Machine Learning to process digital trace data, remote sensing data, to develop maps that can give policymakers a sense of the geographic location of poverty and wealth in the country.

(Jean *et al.*, 2016) states that in order to have reliable Machine Learning is necessary to have a reliable labels, meaning reliable ground true data.
The authors go to most recent household survey data in Nigeria with 42 000 households. In each of those surveys the hours are from 3 to 5 hour face-to-face.

There is a lot of debate about the possible measures for poverty.
One of the standard things is to construct an index, single scalar value, that represents the socioeconomic status of household.

In the case of these household surveys in low and middle income countries probably the most common thing is to construct a wealth index using principal component analysis on a vector of assets and household characteristics.
The x's are going come to satellite imagery.
The idea is that, gather high resolution satellite imagery from the entire country, matching the satellite imagery to the actual geo coordinate of survey household and then extract features from the satellite imagery.
At the end the outcome is a vector of characteristics that are derived from the satellite imagery surrounding the area in which that surveyed household lives.

Afterwards it is possible to apply standard supervised learning methods in particular gradient boosting to predict y from x.
The point is to be able to take an arbitrary point on a map where survey did not happen and then to be able to impute the expected wealth of household in that area using the satellite imagery from that area.

(Chi *et al.*, 2022) have use the supervised neural network to extract features form the images.
Instead of these x features, that describes what is the roofing material of the houses in that image or what is the quality of roads in that image, telling a meaningless characteristics that are really hard to interpret.

Does a model trained and calibrated in Nigeria predict in neighbouring country?

As sort of expect, the models tends to work better in regions where the relationship between visual images and ground truth wells is more constant.

(Chi *et al.*, 2022) produced a map where two square kilometres estimates of poverty for everywhere in a word.
They used 56 different household surveys to train the model but then to evaluate the model the authors compared the estimated from methods trained on households survey data to estimates that come from census data, where census data collection is totally independent.

In addition to the government saying, (Blumenstock, Cadamuro and On, 2015), (Khan and Blumenstock, 2016), (Blumenstock, 2018), (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020), wants to focus on the poorest regions of the country but also in targeting the poorest mobile subscribers.

There it the possibility to identify the owners of a mobile phone, in particular the poorer one rather than wealthier, and transfer to the poorest people rather than everyone in the region.

The idea is that wealthy people use their phones differently from poor people.
With the right training data and applying some basic Machine Learning tools than it is possible to use the patterns of phone use in order to predict wealth and poverty.

From (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020),

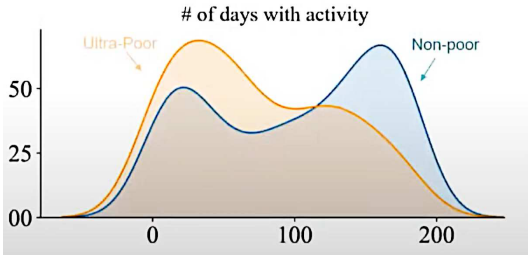

Figure 5: Phone data, Source: (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020)

Figure 5 shows the ability to differentiate ultra-poor households.

Figure 6 point out that looking at the number of unique days in which a head of household use their phone, the distributions are very different.

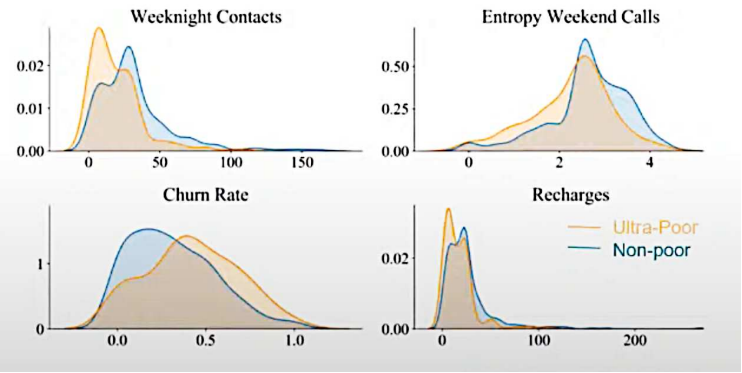

Figure 6: Phone data, Source: (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020)

This is true across all sorts of different metrics of phone use that can be extracted from mobile phone data.
The question now is how well can Machine Learning algorithm use those differences to predict the subscriber's wealth of poverty.

The authors explain that the crux is having a label training dataset in which it is possible to maps mobile phone metadata, quantifying phone use measurements to true measure of wealth.

The approach is conducted using actual surveys with people through the survey process, getting informed consent (taking into account ethical privacy).
In this way it is possible to have the measure of wealth, or consumption in general, socioeconomic status that comes from the survey and then going back to the phone company, getting access to the original call detail records which is the metadata about how people are using their phone.

These data are composed by the name of the person who's making the phone call and who's receives the phone called, the precise hour on a specific day.

Generally geographic information are captured in the phone logs (approximate location of the phone call).

Afterwards there is a particular passage, called “black feature engineering” by which it is possible to go from the raw transaction log data to set of metrics fed into a supervised learning exercised.

The outcome is a fitted model f from which it is possible to predict wealth ($\hat{y}$) that comes the survey from the phone data (x) that comes from the records of mobile phone records.

Once having fitted model f, it is possible to get a predicted wealth score ($\hat{y}$) for anyone who uses phone, not just the people who are part of label survey data set.
Subject concerns of privacy involved in accessing mobile phone metadata.

The “black feature engineering” approach is based on cellular automata deterministic finite automata for people who have $CS101$ as an undergrad, which is a core construct in computer science.

The key of the success of these algorithm is how much the useful variation in the raw data is captured through the process that fed into the supervised learning algorithm.

The features that were showing up in the model are basically the best predictors.

A lot of them have to do with geographic factors, others have to do whether they are making a lot of calls in the evening relative to the daytime.

The accuracy of phone-based prediction in Togo are summarized in Table 2.

From (Aiken *et al.*, 2022a)

	Consumption	PMT	Asset Index
Panel A: 2018-2019 Field Survey (N = 4,171)			
ML	0.46	0.62	0.74
Single Feature	0.13	0.16	0.11
Panel B: 2018-2019 Field Survey, Rural Only (N = 2,306)			
ML	0.31	0.44	0.51
Single Feature	0.09	0.12	0.08
Panel C: 2020 Phone Survey (N = 8,915)			
ML	--	0.41	0.40
Single Feature	--	0.13	0.14

Table 2: Accuracy of phone based predictions in Togo, Source: (Aiken et al., 2022a)

In Table 2 there is the contrasting of the supervised Machine Learning which gives an R^2 of 0,46.

A more intuitive approach: defining a measure in order to quantify how much did the person spend on their phone in the last month.

The authors states that in general people who spend more on their phone tend to spend more in other things than they tend to be wealthier.

But it turns out that while this is a decent predictor of wealth, the proportion of the variation and wealth that is able to explain with this one feature model, is much less than going through this entire Machine Learning passage.

From (Blumenstock, Cadamuro and On, 2015), (Khan and Blumenstock, 2016), (Blumenstock, 2018), (Khan and Blumenstock, 2019), (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020), it is possible to use this approach in different contexts.

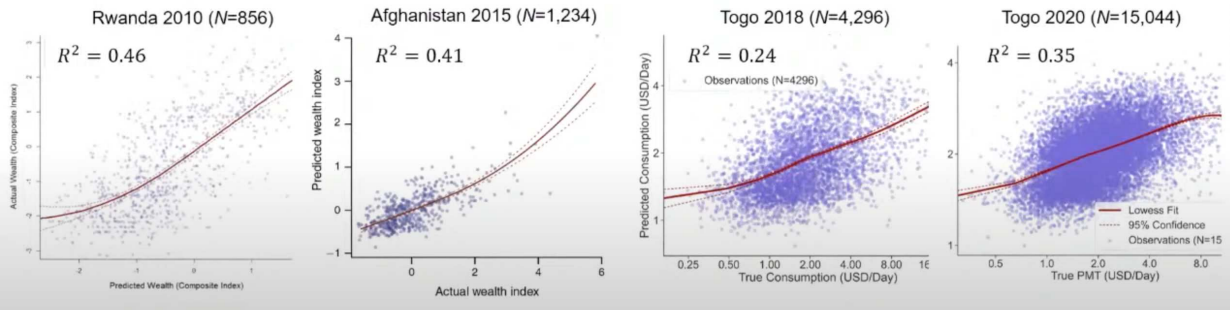

Figure 7: Measuring povery across countries, Source: (Blumenstock, Cadamuro and On, 2015), (Khan and Blumenstock, 2016), (Blumenstock, 2018), (Khan and Blumenstock, 2019), (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020)

There are phone data captures about between a quarter and a half of the variation in socio economic status.

This is not a perfect predictor by any means of socioeconomic status but there is a broad ability to predict.

Then the authors continue to come together this application to targeting in total.

Figure 8: Predicting wealth vs targeting transfers: example, Source: (Blumenstock, Cadamuro and On, 2015), (Khan and Blumenstock, 2016), (Blumenstock, 2018), (Khan and Blumenstock, 2019), (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020)

The idea is looking at the true consumption, the people to the left of this vertical dashed line are the true poor, those of the people that wants to provide benefits to, the other are the non-poor.

Depending on the budget constrain, the dashed line might slide to the left or to the right.
On the y-axis there is the predicted consumption.

If the program is target based on predicted consumption then all of the people before the horizontal line are not going to get paid and all of the people below the horizontal line are.

The confusion matrix that results from this targeting rubric maximize the number of true positives and true negatives and then reduce the number of false positives and the false negatives.

Those measurement can improve the policy.
The evaluation results are going to compare different approaches to targeting benefits within an unconditional cash transfer program, using Togo as a case study, (Aiken *et al.*, 2022b)

1. Geographic, poorest prefectures
2. Geographic targeting which is a very common approach which use geographic information and provide benefits to everyone living in the poorest regions in the country
3. Comparing that to mobile phone based targeting

Key results: how these different methods perform according to technical measurements based on a confusion matrix.

From (Aiken *et al.*, 2022a)

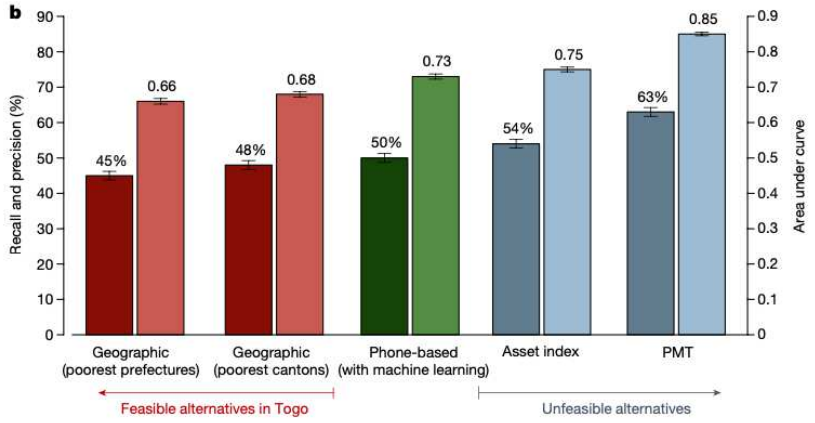

Figure 9: Targeting of the Novissi program in rural areas, Source: (Aiken et al., 2022a)

This evaluation is done use nationally representative household survey.
The process that the authors put in place is training the Machine Learning algorithm, targeting benefits to the poorest people based on the mobile phones.

The outcome: the phone based approach works better than these sort of relatively naïve but feasible alternatives whereas these approach that are not feasible in Togo, because they required a census household survey but that are used in a lot of other programs and in different countries work better than the phone based methods.

It is possible to improve the performance of social welfare.

From (Aiken *et al.*, 2022a)

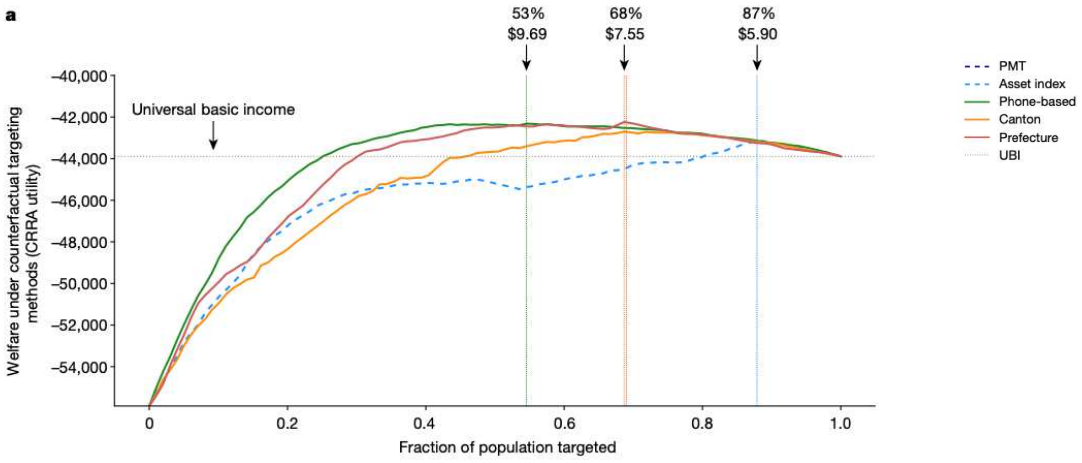

Figure 10: Targeting of the Novissi program in rural areas, Source: (Aiken et al., 2022a)

The necessary assumption about welfare:

- Function of welfare is an CRRA utility function.

$$U = \frac{\sum(y_i + b_i)^{1-\rho}}{1-\rho} \quad (36)$$

- Budget = $ 5 million
- $\rho = 3$
- 5 monthly transfers
- $0.88 fee for each transfer

Figure 10 represents, the area under the curved, that is scaled, by the expected utility of the beneficiary.
The x-axis represents the size of the transfer, as this goes down, there is a fixed budget constraint.
It is possible to target a large portion of the population paying them in a limit of $0.14.

Vertical lines shows the optimal transfer size for that specific targeting mechanism.
The optimal transfer size is close to the actual value that is used in Togo.

The authors continues, computing an analysis in order to understand if Machine Learning can be used to target transfer with respect of other approaches.

Specifically, they make an analysis of the different sources of exclusion in this actual program that use Machine Learning and mobile phones to target transfers.

The sources of exclusion:

- Unregistered people
- People that do not have a SIM card and have access to mobile phone
- People that are do not know that they need to register via this USSD platform
- Targeting errors arise from the Machine Learning algorithm itself. The Machine Learning algorithm is one key source of exclusion errors,

The point is to trying to isolate the people whose predicted consumption is below some threshold through a combination of geographic targeting and phone based targeting.

It is possible to improve the accuracy and the social welfare using people's mobile phone data to target cash transfers taking into account the privacy implications of giving a program administrator access to some amount of private data.

There is a growing literature in private Machine Learning and distributed federated Machine Learning that ask the question what sort of scarify in predictive accuracy need to be made, in order to provide some sort of privacy guarantees on the underlying data.
It is possible to characterize the pareto trade-off between accuracy and privacy but then it is up to the policy maker to decide where along the frontier their policy wants to land.

4.2 Banking and Finance

(Crépon *et al.*, 2015) conducted a randomized experiment in Morocco to measure the impact of micro finance on financial outcomes.

The set up was structured in this way: there are 162 villages with 5000 households, these villages are divided into 81 pairs and then one village is gets to be in a treated group and another one gets to be in the control group.
In the treated village a micro finance institution opens branches and three years later the outcomes are observed.
The Y variable is the financial profit of the household after these three years.
The D is the indicator of being able to access to micro finance services.
The Z, the baseline covariance, includes 22 household characteristics such as number of households members, number of adults the previous use of credit.

The inference part includes the fixed village effects as well as the clustering at the village level.

The authors apply different Machine Learning methods like the Elastic network for predicting baseline scores indicate function, Boosting, Neural net, Random Forest.

The question is how do we select the best Machine Learning estimator for CATE.

$$\Lambda \coloneqq |\beta_2|^2 Var\big(S(Z)\big) \propto Corr^2\big(s_0(Z), S(Z)\big) \tag{37}$$

This equation (37) can be explained in this way.
Λ is equal to the heterogeneity loading parameters squared times the variance of the Machine Learning score, that is going to be proportional to the correlation squared between the true CATE and the Machine Learning proxy.
Selecting the Machine Learning methods gives the highest Λ is equivalent to electing the Machine Learning method that gives us the best Machine Learning proxy for CATE.

Similar device works also for selecting the best Machine Learning tool from the point of view of group average treatment effect analysis.

	Elastic Net	Boosting	Nnet	Random Forest
Profit (Λ)	32307828	17105855	20404000	39286050
Notes: Medians over 1,000 splits.				

Table 3: Choosing Best Machine Learning producing best BLP of CATE, Source: (Chernozhukov *et al.*, 2018)

The result is that the best performing methods are the random forest, which is minimal tuning or no tuning with default parameters often produces the best or near the best results.
Other method is the elastic net, which produces like second best results.
Neural nets do not perform particularly well because the sample size is not large (5.000).

The discussion is then focused on elastic net and random forest results.

	Elastic Net		Random Forest	
	ATE β_1	HET β_2	ATE β_1	HET β_2
Profit	1553	0.244	1603	0.279
	(-1344,4389)	(0.079,0.416)	(-1276.,4536)	(0.046,0.518)
	[0.584]	[0.008]	[0.521]	[0.039]
Median estimates, CIs, and p-values computed over 1000 splits.				

Table 4:BLP of the Effect of Microfinance on Profits, Source: (Chernozhukov *et al.*, 2018)

These numbers represents the averaged treatment effects (ATE), then the confidence intervals and the p-value.
β_2 is the heterogeneity loading parameter.

It is notable that all values are quite different from 1, meaning that there is a little level of correction post processing, but they are also substantially different from zero, this means that there is both heterogeneity and also these Machine Learning tools provide relevant predictors of this heterogeneity.

Given this proxies the authors distinguishes between the most effective group and the least effective group.

	Elastic Net			Random Forest		
	Most Affected γ_5	Least Affected γ_1	Difference $\gamma_5 - \gamma_1$	Most Affected γ_5	Least Affected γ_1	Difference $\gamma_5 - \gamma_1$
Profit	10644.939	-1152.242	**11768**	11540	-2031	**14037**
	(2146,19096)	(-7250,4952)	(1077,22422)	(2965,20955.576)	(-8721,4796)	(2459,25833)
	[0.028]	[1.000]	[0.061]	[0.014]	[1.000]	[0.037]

Table 5: GATEs of Microfinance on Profits, Source: (Chernozhukov *et al.*, 2018)

Looking at the average profit that results from the intervention, it is observable that for the most affected group the point estimate is 10.000 monetary units, the confidence intervals varies from 2.000 to 19.000.

For the least affected group the estimate is essentially zero with wide confidence band.

The difference in the parameters is around 11.000 and it is both economically and statically significant.

A qualitatively similar results are obtained with random forest, the difference is around 14.000 and it is economically and statistically significant.

There appears to be heterogeneity.

The results are analogous to this table could also be shown graphically,

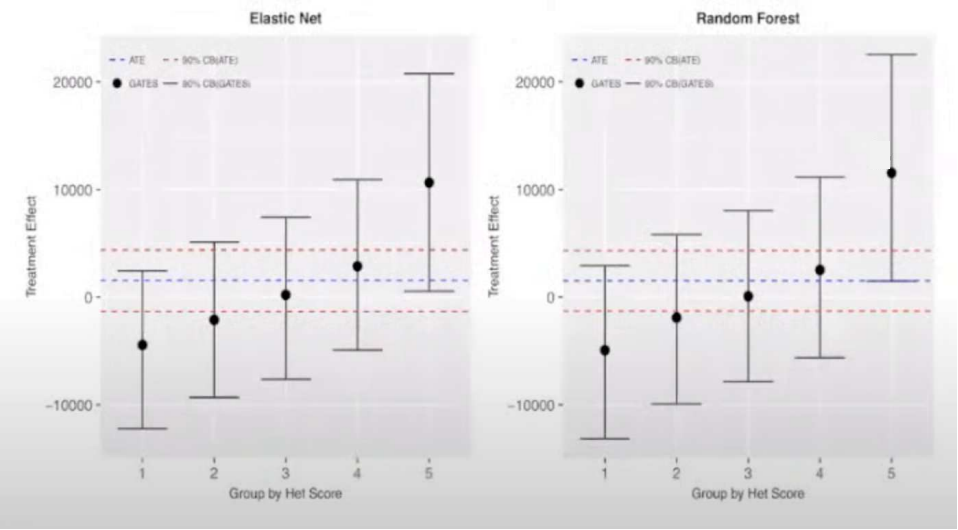

Figure 11: ATE and GATES of Microfinance on Profits, Source: (Chernozhukov *et al.*, 2018)

The groups are divided in five quintiles.

On y-axis there is a group average treatment effect, the black point are the estimates and the confidence bands are constructed to be the 90% and in the end the simultaneous confidence sets.
This graphical way of presenting results it could also be useful to compute different tests like the portman two tests.
For a more precise testing it is needed to go to specific parameters and compute a tailor-made inference.

The most affected and least affect groups, so we are doing a classification analysis for micro finance effects.

	Elastic Net			Random Forest		
	10 % Most	10 % Least	Difference	10 % Most	10 % Least	Difference
Profit						
Head Age	**34.1** (31.2,37.0) -	**40.4** (37.5,43.4) -	**-6.5** (-10.7,-2.5) [0.003]	29.2 (25.7,32.6) -	33.7 (30.390,37.108) -	**-5.8** -10.566,-1.217) [0.029]
Non-agricultural self-emp.	0.181 (0.140,0.222) -	0.108 (0.068,0.149) -	0.082 (0.022,0.138) [0.014]	0.153 (0.113,0.192) -	0.099 (0.058,0.139) -	0.051 (-0.003,0.105) [0.129]
Borrowed from Any Source	**0.180** (0.130,0.230) -	**0.257** (0.207,0.307) -	**-0.091** (-0.160,-0.022) [0.020]	0.144 (0.098,0.190) -	0.162 (0.122,0.206) -	-0.032 (-0.095,0.029) [0.578]

Table 6: GATES of Microfinance on Profits, Source: (Chernozhukov *et al.*, 2018)

Looking at the results, the most affected group tends to be younger household with less borrowing experience. This opening micro finance institution seems to help younger households with maybe less borrowing experience or more limited access to other financial resources.

This paper try to proposed a generic assumption-free strategies to make inference on key features of heterogeneous effects and randomized experiments.
The key features include the best linear predictor, the group average treatment effect and classification analysis (average characteristics of the most missed affected group).

They also perform a repeated data splitting to avoid overfitting, the inference quantifies the uncertainty coming both from the parameter estimation and from the data splitting.

Data splitting of course creates a loss of power.
The standard errors are scaled by $\frac{1}{\sqrt{N}}$ thus the precision is reduced by a factor of $\frac{1}{\sqrt{2}}$.

Computing a sample splitting, allows to protect against overfitting and gaining certain robustness.
There is trade-offs between the robustness and the data splitting method.

4.3 Politics and Policy

Machine learning can also support the governance of a state in particular can improve the anti-corruption policy using prediction tools.

Anti-corruption entails over 3.6 trillion USD annually expenditure.
(Ash, Galletta and Giommoni, 2021) conducted an analysis in the context of Brazilian municipalities from 2001 to 2012.
In particular, the authors analyses how politicians distributes resources by using Brazilian government auditors.
Brazil has 5563 municipalities within 26 states with a high level of autonomy in budgeting decisions.
During the time of the analysis the state's policy changed in order to creates boundaries to corruption.
Governments implemented a system with a randomly selected audited whose task was to assure the fairness the allocation of funds.

Auditors provides a set of information which were then used by the authors in order to create a sort of indices for predicting corruption at municipal level.
Corruption is interpretable from the authors as "narrow corruption" such as fraud, favouritisms and illegal procurement.

In order to test the robustness of the analysis, the authors compute an alternative measure for corruption mechanism, using different documents and for different set of audits.

The prediction process is done using information from the municipal budget in order to compute the probability of the corruption in a given municipality.
This estimates is done using Machine Learning with a specific tool, Gradient Boosting Model.

The approach is to conduct an analysis of budget predictors and corruption outcomes, where x variables are the budget features while y can takes two values, one if in the years of the analysis the audits found corruption, zero instead in which audits did not find corruption.

For the Machine Learning procedure the regression implemented tries to identify the municipality corruption based on observable budget features.

The analysis is done using different regression metrics, both baseline models, like the OLS methods, but also penalized models, like Lasso and Logistic.

The main difference between these two methods is that the latter have inside the regression a penalized parameter (called λ) that helps to put much weights on the data which have large coefficients.
This way, regression is improved by avoiding overfitting.

All these methods are then compared with Gradient Boosting model, specifically Gradient Boosting trees.
This method compute an iterative decision trees splitting the node depending on the value decides (for example $x > 100$) until reaching the terminal node with the associated prediction. Then cross-validation is developed and included L1 (Lasso) and L2 (Ridge) regularization penalties (which counts the non-zero parameters), the learning rate parameter, the magnitude of the trees and the limit for the last nodes.

Table 7 compared the different results from XGBoost, baseline methods and penalized regression.

The first one is Guessing and represents the case in which there is no corruption (called from the authors " modal category").

	Guessing (1)	OLS (2)	Lasso (3)	Logistic (4)	XGBoost (5)
Accuracy	0.580	0.476	0.474	0.560	0.723
		(0.022)	(0.022)	(0.022)	(0.012)
AUC-ROC		0.487	0.507	0.568	0.777
		(0.016)	(0.012)	(0.016)	(0.013)
F1	0.000	0.480	0.538	0.545	0.632
		(0.031)	(0.050)	(0.054)	(0.018)

Table 7: "Out-of-Sample Metrics for Predicting Corruption", Source: (Ash, Galletta and Giommoni, 2021)

Notably, in the table it is summarized the metrics that are out-of-sample in order to predict the corruption.

Accuracy explained how model approximately the truth well.

AUC-ROC model is the "Area under the receiver operator characteristic curve" which ranges from 0.5 to 1.

It can be reading as the probability that a corrupted municipality is classified with a higher grade, in a randomized way, than the probability of corruption with respect to a non-corrupted municipalities, in an arbitrarily way.

Since the AUC-ROC requires a classification based on grades, for the Guessing model there is no a defined value.

Finally, the authors defined F1 as "proportion true corrupt within the set predicted corrupt" and recall "proportion predicted corrupt within the set true corrupt".

F1 models penalized the false positive and the false negatives.

Each cells characterized the mean and standard deviation in parathesis, computed as the average test-set estimation (ATE) for each model.

It is ascertain that the model which better performed in terms of the three performance metrics is the Gradient Boosting trees with an average test set as 0.723, 0.777, 0.632, respectively.

The authors continue calibrating each predicted class with the correspondent true corruption rate.

This analysis is reported in:

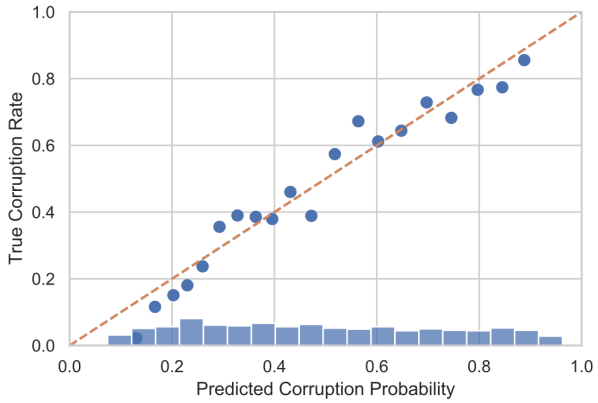

Figure 12: "Predicted Probabilities in Test Set are Well-Calibrated to True Corruption Rates", Source: (Ash, Galletta and Giommoni, 2021)

Figure 12 shows how the model is well fine-tuned.
The predicted corruption probability is closed to the perfect calibration line, so to the true corruption rate.

(Ash, Galletta and Giommoni, 2021) analysed how much a surprising increase of public revenues, defined as federal transfer, affect corruption.
authors specified that the federal transfers are the largest cause of earnings.

Table 8 represents the outcome from the regression analysis, where the dependent variable is the corruption and the other variables represents external variables which depends on population.

	Audited cities (1)	All cities (2)	Non-audited cities (3)
Panel A. First Stage			
Theoretical transfers	0.6805***	0.6909***	0.6996***
	(0.0205)	(0.0233)	(0.0230)
Panel B. Reduced Form			
Theoretical transfers	0.0040***	0.0041***	0.0040***
	(0.0009)	(0.0003)	(0.0003)
Panel C. 2SLS			
Actual transfers	0.0058***	0.0059***	0.0057***
	(0.0013)	(0.0005)	(0.0005)
N. Observations	1115	5808	4693

Table 8: "Effect of Revenue Shocks on (Predicted) Corruption", Source: (Ash, Galletta and Giommoni, 2021)

Audit cities represents municipalities that collected an audit, Non-audited cities represents municipalities that did not collect an audit and finally all cities.
It is necessary to specify that for the Panel A the dependent variable is "actual transfer", for Panel B it is "predicted corruption", while in Panel C the dependent variable corresponds to "predicted corruption and actual transfer is instrumented with theoretical transfers".

It is observable how regressions are all statistically significant and how varying from different estimators, there is no altering in terms of the size of coefficients.

The analysis continues by assessing the effects of a control on corruption.

$$y_{ist} = \sum_{k=-3, k\neq -1}^{5} \beta_k D_{ist}^k + \delta_i + \lambda_t + W_{ist}'\phi + \epsilon_{ist} \quad (38)$$

Where y is the annual corruption prediction of i, refers to municipality, s states and t year.
D, instead, represents the dummy variable for the years before and after the control.
δ is the municipality fixed effect, λ is year fixed effect, W are others controls.

The estimates are reported in the following table:

	All cities (1)	Cities with corruption (2)	Cities without corruption (3)
Year pre4 and behind	-0.0171 (0.0245)	-0.0287 (0.0427)	-0.0052 (0.0748)
Year pre3	-0.0118 (0.0190)	-0.0024 (0.0287)	-0.0164 (0.0476)
Year pre2	-0.0078 (0.0124)	0.0203 (0.0205)	-0.0390 (0.0302)
Audit year	-0.0358*** (0.0109)	-0.0177 (0.0145)	-0.0506* (0.0254)
Year post1	-0.0429** (0.0166)	-0.1002*** (0.0200)	-0.0597 (0.0387)
Year post2	-0.0238 (0.0246)	-0.1456*** (0.0311)	0.0205 (0.0545)
Year post3	-0.0253 (0.0262)	-0.1924*** (0.0376)	0.0659 (0.0672)
Year post4	-0.0276 (0.0308)	-0.2307*** (0.0490)	0.0903 (0.1018)
Year post5	-0.0156 (0.0418)	-0.2585*** (0.0620)	0.1581 (0.1185)
Years post6 and more	-0.0364 (0.0478)	-0.3260*** (0.0711)	0.1756 (0.1294)
N. Observations	17252	8895	3086
Adjusted R^2	0.535	0.510	0.538

Table 9: The effect of an audit, Source: (Ash, Galletta and Giommoni, 2021)

While graphically are represented:

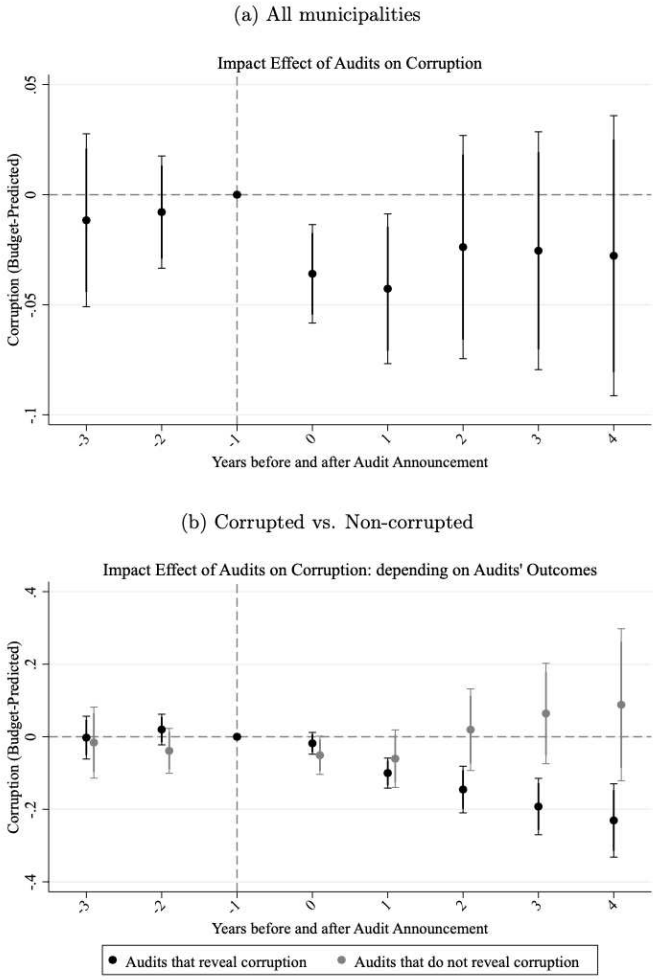

Figure 13: The effect of an audit, Source: (Ash, Galletta and Giommoni, 2021)

It is noticeable, in Panel A, how after the control ($k = 0$) there is a decrease in the prediction of corruption.

Panel B, instead, documents the incident investigation effects for audits where clear corruption is detected (black dots) and for those where corruption is not detected (grey dots). Sixteen of these trends are quite different. When corruption is detected (black dots), the negative effect is much larger, ranging from -1.7% to -25.8%, which is significant compared to the treatment average of 55.8%. On the contrary, if the audit does not detect corruption or irregularities (grey points), it has no effect.

In conclusion this paper shows the potential prediction of corruption using public data.
The resulting composite measures can then be used in empirical analysis, to produce similar empirical results using estimates of corruption in municipalities that have never been audited. Optimistically, in the future, these techniques will be useful to researchers to compute further analysis on other variables.

5 Conclusions

Building the algorithm is not the hard part, and this can be visible in other things applied navigating these micro-econometric issues.
These issues are largely more than Machine Learning.
The interesting part of being in this filed in this moment, is not just do the mechanical applications, it is to be live to these type of complexities.

These complexities, like selective labels, omitted variable, make it a non-natural fit to apply Machine Learning.
They are what create both econometrically interesting questions to resolve and also applied questions and work that need to be done.

There are many more problems like this, where the core problem is predicting.
Predictive problem, as described in this analysis, require clear goal, that is represented in the data and but it also required individualization.

Understanding the problem applying to and asking does not really fit a very careful predictive model is key.

The application of Machine Learning explained by (L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, 2020), on how it is possible to use phone data to better find who's poor and target poverty, is an example of implementation.

Another problem arise when people are unemployed, in particular predicting for how long a person will remain unemployed.

There is a tons of problems in terms of finance, where every person wants to know when they are buying a house, if they can afford it and about the eviction rate late payment default.

In education, there are a lot of work that looks at the experimental effect of an additional teacher, which is a $\hat{\beta}$ problem.
Holding that constant, how do you know who makes a good teacher?
This is a prediction about the performance of the teacher or performance of the students under that teacher.

What is happening now is that we are starting to see many more data-driven decisions and Machine Learning can change that huge part of an unexplored data of the space of predictive decisions.

Economic tools can also improve Machine Learning, this is not a one-way street.
The real case application describes in the last chapter of this thesis, are examples.

Prediction can aid in economic theorizing and Machine learning gives us a very new way of working with science.

Prediction is an inherently important scientific tools, it differentiates from causal inference, it could be possible to say that it is the complementary part.

Machine Learning can help us computing a causal inference in a better way, but whenever a new tool comes along, people first use it to solve old problems, but eventually the big gains do not come from the new tools solving old problems, it comes from the new tool helping us solve new problems.

Indeed, changing the entire scope of the discipline this happen with behavioural economics, with causal imprints and Machine Learning is going to totally change what kind of question do we event tackle.

References

Aiken, E. *et al.* (2022a) 'Machine learning and phone data can improve targeting of humanitarian aid', *Nature*. Springer US, 603(7903), pp. 864–870. doi: 10.1038/s41586-022-04484-9.

Aiken, E. *et al.* (2022b) 'Machine learning and phone data can improve targeting of humanitarian aid', *Nature*, 603(7903), pp. 864–870. doi: 10.1038/s41586-022-04484-9.

Ash, E., Galletta, S. and Giommoni, T. (2021) 'A Machine Learning Approach to Analyze and Support Anti-Corruption Policy', *SSRN Electronic Journal*, 2021(June 2021). doi: 10.2139/ssrn.3830220.

Athey, S. (2018) 'The impact of Machine Learning on Economics'.

Athey, S. and Imbens, G. W. (2019) 'Machine Learning Methods That Economists Should Know about', *Annual Review of Economics*, 11, pp. 685–725. doi: 10.1146/annurev-economics-080217-053433.

Bharadwaj, Prakash, K. B. and Kanagachidambaresan, G. R. (2021) *Pattern Recognition and Machine Learning*, *EAI/Springer Innovations in Communication and Computing*. doi: 10.1007/978-3-030-57077-4_11.

Bickel, P. *et al.* (1998) *Springer Series in Statistics Perspectives in Statistics*.

Blumenstoc, J. (2021) 'Machine Learning in Economics Summer Institute 2021', in.

Blumenstock, J., Cadamuro, G. and On, R. (2015) 'Predicting poverty and wealth from mobile phone metadata', *Science*, 350(6264), pp. 1073–1076. doi: 10.1126/science.aac4420.

Blumenstock, J. E. (2018) 'Estimating Economic Characteristics with Phone Data', *AEA Papers and Proceedings*, 108, pp. 72–76. doi: 10.1257/pandp.20181033.

Breiman, L. (1996) 'Bagging Predictors'.

Breiman, L. (2001) ‘Random Forests’.

Chernozhukov, V. *et al.* (2018) ‘GENERIC MACHINE LEARNING INFERENCE ON HETEROGENEOUS TREATMENT EFFECTS IN RANDOMIZED EXPERIMENTS, WITH AN APPLICATION TO IMMUNIZATION IN INDIA’.

Chi, G. *et al.* (2022) ‘Microestimates of wealth for all low- and middle-income countries’, *Proceedings of the National Academy of Sciences of the United States of America*, 119(3), pp. 1–11. doi: 10.1073/pnas.2113658119.

Crépon, B. *et al.* (2015) ‘Estimating the impact of microcredit on those who take it up: Evidence from a randomized experiment in Morocco’, *American Economic Journal: Applied Economics*, 7(1), pp. 123–150. doi: 10.1257/app.20130535.

D1Etterich, T. (1995) ‘Overfitting and Undercomputing in Machine Learning’, *ACM Computing Surveys (CSUR)*, 27(3), pp. 326–327. doi: 10.1145/212094.212114.

Delua (2021) *Supervised vs Unsupervised Learning*.

Dreiseitl, S. and Ohno-Machado, L. (2002) ‘Logistic regression and artificial neural network classification models: A methodology review’, *Journal of Biomedical Informatics*, 35(5–6), pp. 352–359. doi: 10.1016/S1532-0464(03)00034-0.

Gammerman, A., Vovk, V. and Vapnik, V. (2013) ‘Learning by Transduction’, (l), pp. 148–155. Available at: http://arxiv.org/abs/1301.7375.

Gillis, T. B. and L, J. S. (2019) ‘Big Data and Discrimination’, *University of Chicago Law Review*, 86(2), pp. 459–487.

Grech, V. and Calleja, N. (2018) ‘WASP (Write a Scientific Paper): Parametric vs. non-parametric tests’, *Early Human Development*. Elsevier, 123(xxxx), pp. 48–49. doi: 10.1016/j.earlhumdev.2018.04.014.

Hastie, Trevor; Tibshirani, Robert; Friedman, J. (2013) *The Elements of Statistical Learning Data Mining, Inference, and Prediction Second edition*.

Jean, N. *et al.* (2016) 'Combining satellite imagery and machine learning to predict poverty', *Science*, 353(6301), pp. 790–794. doi: 10.1126/science.aaf7894.

Jerven, M. (2013) *Poor Numbers: How We Are Misled by African Development Statistics and What to Do about It*.

Kaur, A. and Kumar, R. (2015) 'Comparative Analysis of Parametric and Non-Parametric Tests', *Journal of Computer and Mathematical Sciences*, 6(6), pp. 336–342.

Khan, M. R. and Blumenstock, J. E. (2016) 'Predictors without borders: Behavioral modeling of product adoption in three developing countries', *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu, pp. 145–154. doi: 10.1145/2939672.2939710.

Khan, M. R. and Blumenstock, J. E. (2019) 'Multi-GCN: Graph convolutional networks for multi-view networks, with applications to global poverty', *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 606–613. doi: 10.1609/aaai.v33i01.3301606.

Knight, K. and Fu, W. (2000) 'Asymptotics for Lasso-type estimators', *Annals of Statistics*, 28(5), pp. 1356–1378. doi: 10.1214/aos/1015957397.

L. Aiken, Emily; Bedoya, Guadalupe; Coville, Aidan; E. Blumenstock, J. (2020) 'Targeting Development Aid with Machine Learning and Mobile Phone Data: Evidence from an Anti-Poverty Intervention in Afghanistan'.

Lakner, C. *et al.* (2022) 'How much does reducing inequality matter for global poverty?', *Journal of Economic Inequality*. The Journal of Economic Inequality, 20(3), pp. 559–585. doi: 10.1007/s10888-021-09510-w.

Leeb, H. and Pötscher, B. M. (2008) 'Can one estimate the unconditional distribution of post-model-selection estimators?', *Econometric Theory*, 24(2), pp. 338–376. doi: 10.1017/S0266466608080158.

Loh, W. Y. (2011) ‘Classification and regression trees’, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1), pp. 14–23. doi: 10.1002/widm.8.

McDonald, G. C. (2009) ‘Ridge regression’, *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), pp. 93–100. doi: 10.1002/wics.14.

Mullainathan, S. and Spiess, J. (2017) ‘Machine learning: An applied econometric approach’, *Journal of Economic Perspectives*, 31(2), pp. 87–106. doi: 10.1257/jep.31.2.87.

Newey, W. K. and Robins, J. R. (2018) ‘Cross-Fitting and Fast Remainder Rates for Semiparametric Estimation’, pp. 1–43. Available at: http://arxiv.org/abs/1801.09138.

Schapire, R. E. (2013) *Boosting: Foundations and Algorithms*, *Kybernetes*. doi: 10.1108/03684921311295547.

Schierle, Martin; Schulz, S. (2007) ‘Bootstrapping algorithms for an application in the automotive domain’.

Shalev-Shwartz, S. and Ben-David, S. (2013) *Understanding machine learning: From theory to algorithms*, *Understanding Machine Learning: From Theory to Algorithms*. doi: 10.1017/CBO9781107298019.

Shankhar, B. S. (2020) *Structuring your Machine Learning projects*.

Suthaharan, S. (2016) *Decision Tree Learning*. doi: 10.1007/978-1-4899-7641-3_10.

Tibshirani, R. (1996) ‘Regression Shrinkage and Selection via the Lasso’.

Varian, H. R. (2014) ‘Big data: New tricks for econometrics’, *Journal of Economic Perspectives*, 28(2), pp. 3–28. doi: 10.1257/jep.28.2.3.

Wb, U. G. *et al.* (2020) ‘Social Protection and Jobs Responses to COVID-19 : A Real-Time Review of Country Measures’, 10.

Yadav, S. and Shukla, S. (2016) 'Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification', *Proceedings - 6th International Advanced Computing Conference, IACC 2016*. IEEE, (Cv), pp. 78–83. doi: 10.1109/IACC.2016.25.

Yeh, C. *et al.* (2020) 'Using publicly available satellite imagery and deep learning to understand economic well-being in Africa', *Nature Communications*. Springer US, 11(1), pp. 1–11. doi: 10.1038/s41467-020-16185-w.

Zeng, X. D., Chao, S. and Wong, F. (2010) 'Optimization of bagging classifiers based on SBCB algorithm', *2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010*. IEEE, 1(July), pp. 262–267. doi: 10.1109/ICMLC.2010.5581054.

Zhang, C. H. and Huang, J. (2008) 'The sparsity and bias of the lasso selection in high-dimensional linear regression', *Annals of Statistics*, 36(4), pp. 1567–1594. doi: 10.1214/07-AOS520.

Zhao, P. and Yu, B. (2006) 'On model selection consistency of Lasso', *Journal of Machine Learning Research*, 7, pp. 2541–2563.