Interpretability

Lipton, in the paper titled "*The Mythos of Model Interpretability*" talks at length about what is interpretability and why it is important. To begin with, the author mentions five objectives that interpretations serve but which are not formally defined. One of these objectives that the paper talks about is trust. The author says, "If the model tends to make mistakes in regions of input space where humans also make mistakes, … then it may be considered trustworthy in the sense that there is no expected cost of relinquishing control". I do not agree with this. Consider the case of self-driving cars. One of the main purposes of an autonomous vehicle is to provide a better and safer driving experience. However, if the autonomous car also causes accidents in scenarios where humans are bound to err, it can neither be called accurate nor trustworthy.

Further, the author correctly points out that linear models are not strictly more interpretable than deep learning models, with respect to algorithmic transparency. To elaborate, consider a housing price prediction model. These models are generally based on hedonic pricing methods, i.e., a good (in this case, house) is considered as a sum of individual components (attributes such locality, number of bedrooms, elevator, proximity to schools etc.) which cannot be sold separately in the market. Consider a situation where, keeping everything else constant, the only changing variable is the number of bedrooms. Irrespective of whether it is a linear model or a deep learning model, one can easily interpret that the cost of a 3B house would be greater than that of a 2B house. Next, consider a linear model which has all relevant attributes and transformations of these attributes, for instance, a PCA transformation. One cannot, with certainty, interpret all the components of PCA. This is in line with what the author mentions as loss of simulatability or decomposability of linear models.

Additionally, the paper rightly points out that post-hoc interpretations can be misleading. It is an interesting thought that we, as humans, are known to (un)intentionally optimize an algorithm to present misleading but plausible explanations. An example of this behavior from research context is "P-hacking". P-hacking is the method of discovering patterns which are presented as statistically significant, when, there is no underlying effect. This is a common practice in clinical trials and scientific research. To avoid p-hacking, guidelines such as SPIRIT and ICH-E9 now require all researchers to pre-specify the planned analysis approach in the trial protocol. However, there are loopholes to the pre-specification method as well, but those are out of the scope of this response.

On similar lines, Burrell, in his paper, "*How the Machine Thinks…*" explains that opacity (or non-transparency) is an issue for algorithms deployed in a social context. He describes three forms of opacity – as a competitive advantage, as technical illiteracy, and as the way algorithms operate on large scale. A proposed solution for the first form of opacity is to "make code available for scrutiny, through regulators". However, this solution merely shifts the burden of decision making to regulators rather than the algorithm itself. It makes sense in situations that require specialized knowledge such as medical fields, or financial markets. But in case of college applications, there is a higher chance of human bias coming into play. Moreover, a code hardly reflects what a machine learning or deep learning model "learns" from the training data.

All in all, although the right to explanation is one of the important sections under EU's GDPR, it seems like there is a long way until an algorithm itself can be interpretable and consequently, explainable.