Data
Governance
Network

March 2020

Working Paper 07

Operationalizing algorithmic explainability in the context of risk profiling done by robo financial advisory apps

Sanjana Krishnan, Sahil Deo, and Neha Sontakke

About the Data Governance Network

The Data Governance Network (DGN) is a multi-stakeholder community of researchers tackling India's next policy frontiers: data-enabled policymaking and the digital economy. Some of India's leading policy think-tanks – IDFC Institute, National Institute of Public Finance and Policy (NIPFP), Internet Democracy project (IDP) and IT for Change (ITfC) are research nodes of DGN. IDFC Institute also functions as the secretariat of DGN. DGN works to cultivate and communicate research stemming from diverse viewpoints on market regulation, information privacy and digital rights, in an attempt to generate balanced and networked perspectives on data governance—thereby helping governments make smart policy choices which advance the empowerment and protection of individuals in today's data-rich environment.

About CPC Analytics

CPC Analytics is a data-driven public policy and development sector consulting firm based in Berlin and Pune. Bringing together qualitative and quantitative research methods, CPC's team of economists, social scientists, and software engineers support public sector organizations and businesses to undertake policy research and collect, analyse and visualize data. Previous clients include the WHO, UNAIDS, GIZ, GOAL Global, The Graduate Institute in Geneva, and the European Research Centre for Anti-Corruption and State-Building (ERCAS) as well as Siemens, Mitsubishi Materials, and Veganz.

CPC Analytics is currently researching methods to mitigating the negative externalities of AI and algorithmic decision making systems and understand their governance.

Terms of Use

Disclaimer

The views and opinions expressed in this paper are those of the authors and do not necessarily represent those of the organization.

Suggested Citation

Krishnan S., Deo S., & Sontakke N. (2020, January 30). Operationalizing algorithmic explainability in the context of risk profiling done by robo financial advisory apps.

Supported by a grant from Omidyar Network

Abstract

Robo Advisors are financial advisory apps that profile users into risk classes before providing financial advice. This risk profiling of users is of functional importance and is legally mandatory. Irregularities at this primary step will lead to incorrect recommendations for the users. Further, lack of transparency and explanations for these automated decisions makes it tougher for users and regulators to understand the rationale behind the advice given by these apps, leading to a trust deficit. Regulators monitor this profiling but possess no independent toolkit to "demystify" the black box or adequately explain the decision-making process of the robo financial advisor.

Our paper proposes an approach towards developing a 'RegTech tool' that can explain the robo advisors decision making. We use machine learning models to reverse engineer the importance of features in the black-box algorithm used by the robo advisor for risk profiling and provide three levels of explanation. First, we find the importance of inputs used in the risk profiling algorithm. Second, we infer relationships between inputs and with the assigned risk classes. Third, we allow regulators to explain decisions for any given user profile, in order to 'spot check' a random data point. With these three explanation methods, we provide regulators, who lack the technical knowledge to understand algorithmic decisions, a method to understand it and ensure that the risk-profiling done by robo advisory applications comply with the regulations they are subjected to.

Keywords: Algorithmic decision-making systems (ADS), algorithmic regulation, algorithmic explainability and transparency, robo financial advisory apps, fintech, explainable AI

Table of Contents

1. Introduction

There is a growing ubiquity of decision-making algorithms that affect our lives and the choices we make. These algorithms curate our internet and social media feed, trade in the stock market, assess risk in banking, fintech and insurance, diagnose health ailments, predict crime prevention, and a lot more. Broadly, these are known as Algorithmic Decision-making Systems (ADS). Machine learning algorithms are one of the crucial components of ADS and artificial intelligence (AI), and power the automated, independent decision making done by computers. Machines 'learn' by going through millions of data points and find associations and patterns in them. They then apply the learnt rules on new data to predict the outcomes. These algorithms have promised and delivered considerable gains in efficiency, economic growth, and have transformed the way we consume goods, services, and information.

However, along with the gains, these algorithms also pose threats. Several cases have come to light where algorithm powered decisions have given rise to undesirable consequences. An automated hiring tool used by Amazon discriminated heavily against women applying for software development jobs, because the machines learn from past data which has a disproportionate number of men in software positions (Dastin, 2018). Software used for crime prediction in the United States showed a machine bias against African-Americans, exacerbating the systemic bias in the racial composition of prisons (ProPublica, 2016). Google's online advertising system displayed ads for high-income jobs to men much more often than it did to women (Datta, Tschantz, & Datta, 2015). Social media algorithms are found to inadvertently promote extremist ideology (Costello, Hawdon, Ratliff, & Grantham, 2016) and affecting election results (Baer, 2019). Recently, researchers found that racial bias in the US health algorithms reduced the number of Black patients identified for extra care by more than half (Obermeyer, Powers, Vogeli, & Mullainathan, 2019) (Kari, 2019).

In effect, contrary to the promise of unbiased and objective decision making, these examples point to a tendency of algorithms to unintentionally learn and reinforce undesired and non-obvious biases, thus creating a trust deficit. This arises mainly because several of these algorithms are not adequately tested for bias and are not subjected to external due-diligence. The complexity and opacity in the algorithms decision-making process and the esoteric nature of programming denies those affected by it access to explore the rights-based concerns posed by algorithms.

However, if these algorithms make decisions in the public sphere that affect an individual's access to services and opportunities, they need to be scrutinized. Over the last two years, there is a growing call to assess algorithms for concepts like fairness, accountability, transparency, and explainability and there has been an increase in research efforts in this direction.

Our research is situated in this context and we attempt to operationalize the concept of **explainability** in automated tools used in fintech. We have selected the case of **robo financial advisory apps** which conduct a **risk profiling** of users based on a questionnaire and gives users customized investment advice.

What are robo financial advisors?

Robo advisory applications are automated web-based investment advisory algorithms that estimate the best plans for trading, investment, portfolio rebalancing, or tax saving, for each individual as per their requirements and preferences. Typically, a user fills in questionnaire or survey and is classified in either three or five risk classes (ranging from 'low risk' to 'high risk'). Robo advisors open up the potential for finance to be democratized by reducing the financial barrier to entry and providing equal access to financial advice through their low-cost business model (Laboure & Braunstein, 2017).

The first robo financial advisory app was launched in 2008, and the use of such applications has increased with the growth of internet-based technology and the sophistication of functionalities and analytics (Abraham, Schmukler, & Tessada, 2019) (Narayanan, 2016). In a 2014 report, the International Organization of Securities Commission (IOSCO) made a comprehensive effort to understand how investment intermediaries use automated advisory tools. They identified a spectrum of ‘Internet-based automated investment selection tools’ and classified them based on the complexity of the advice that it gives, from a basic level of risk classification to a complex assessment of the customers age, financial condition, risk tolerance, and capacity, among others, to offer automated advice suited to the users investment goals. The output is often a set of recommendations for allocations based on parameters like the size of funds (small, mid-cap), the type of investment (debt and equity funds), and even a list of securities or portfolios (IOSCO, 2014).

This **risk profiling** done by these robo-financial advisors is a crucial step to determine the risk class of the user which determines the investment advice. Irregularities at this primary step will lead to incorrect recommendations for the users. Moreover, unlike human advisors, robo advisors provide no reasons or explanations for their decisions, and this shortcoming reduces the trust that users repose in their advice (Maurell, 2019).

Several robo financial advisory applications operate in India. Prominent ones include PayTM money, GoalWise, Artha-Yantra, Upwardly, Kuvera, Scripbox, MobiKwick, and Tavaga, among others.

1.1. Regulating ADS

(Citron & Pasquale, 2014) argue that transparency and opening the black-box are crucial first steps and that oversight over algorithms should be a critical aim of the legal system. They argue for procedural regularity in assessing all publicly used algorithms to ensure fairness.

The European Union General Data Protection Regulation (EU GDPR) adopted in 2016 lays down comprehensive guidelines for collecting, storing, and using personal data. While it is mainly aimed at protecting data, Article 22 speaks about “Automated individual decision making, including profiling”, specifying that *“data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her”* (subject to exceptions for contract enforcement, law and consent).

It calls for consent, safeguarding the rights and freedoms, and further gives the subject the right to obtain human intervention, express their point of view and contest the decision (EU GDPR, 2016).

(Goodman & Flaxman, 2017) in their review of Article 22 of the GDPR reflect that this could necessitate a 'complete overhaul of widely used algorithmic techniques'. They look at this provision as a 'right to non-discrimination' and a 'right to explanation' when read with other articles in the GDPR. Contrary to this, (Wachter, Mittelstadt, & Floridi, 2016) argue that while the 'right to explanation' is viewed as an ideal mechanism to enhance the accountability and transparency of automated decision-making, there is doubt about the legal existence and feasibility of such a right in the GDPR, owing to the lack of explicit, well-defined rights and imprecise language. They contest that Articles 13-15 of the GDPR merely mandates that data subjects receive 'meaningfully, but properly limited information', what they call the 'right to be informed'. They raise the need for a meaningful right to explanation to be added to Article 22, where data controllers need to give the rationale for decisions, evidence for the weighing of features and logic of decision making.

In the Indian context, (Kapur & Khosla, 2019) observe that dealing with new technologies is one of the most demanding challenges facing regulatory design. (Padmanabhan & Rastogi, 2019) identify that the point of threat to individual and group rights has shifted from data gathering to data processing, and that the regulation of algorithms is unaddressed. Further, they note that there are no clear substantive safeguards against potential harm to social and individual rights, or regulatory mechanisms to mitigate against them in India.

The regulations or governance of algorithms could be cross-sectoral or/and sector specific. A cross sectoral algorithmic governance could imply having a special regulatory or supervisory agency to audit algorithms and oversee its functioning. Calls have been made to establish for a FDA for Algorithms (Tutt, 2016), Machine Intelligence committee (Mulgan, 2016), an AI Watchdog (Sample, 2017), or a Algorithmic Safety Board akin to the US National Transportation Safety Board (Shneiderman, 2017). Such bodies operating at a jurisdictional level would have the power to license algorithms, monitor their use, and investigate them. Additionally, In addition to cross-sectoral regulations (or in the absence of it), sector-specific algorithmic regulations could operate. Regulators in different sectors like healthcare, finance or education can design rules and oversee the working of algorithms that operate in their sector. Given sector-specific challenges and algorithmic use cases, an overarching regulator might not have the capacity, time or domain-knowledge to address the issues (Andrews, 2017), and it may be inappropriate to solely apply solutions across sectors (New & Castro, 2018).

A crucial debate on the regulations is about the capacity of the regulators to deal with the ever-evolving nature and growing ubiquity of technology. The use of technology and algorithms are cutting across sectors and are increasingly used in finance, health, education, mobility, and more. To regulate rapidly transforming sectors, there has been a growing call for the use of RegTech. RegTech (or regulatory technology) are 'technological solutions to regulatory problems' (Chazot, 2015) that use technology for regulatory monitoring, reporting and compliance (Arner, Barberis, & Buckley, 2016). RegTech can use various technical, mathematical and statistical functions to detect financial fraud, biased practices, anti-trust activity, etc. The can also be implemented as tools using which regulators get automated compliance reports, allowing them to monitor tech without understanding its full working, enable cost savings and gain superior monitoring ability. (Arner, Barberis, & Buckley, 2016) refer to this as 'the early signs of real-time and proportionate regulatory regimes'.

1.2. SEBI guidelines for robo advisory tools

While there are no overarching regulations on algorithms in India, some sectoral regulators have delineated guidelines and regulations on use of algorithms in their sectors. Automated tools used in fintech are subject to regulations by the Securities Exchange Board of India (SEBI), a statutory body that regulates the securities market in India. In 2016, they released a consultation paper in which they lay down rules for 'Online Investment Advisory and automated tools' (SEBI, Consultation Paper on Amendments/Clarifications to the SEBI (Investment Advisers) Regulations, 2013, 2016).

In this section, they clearly state that automated tools need to comply with all rules under the SEBI (Investment Advisers) Regulations, 2013, over and above which they are subject to additional compliances

One primary function of an investment advisor under the Investment Advisors Regulations is to profile the risk class of the user. The Investment Advisors regulations states that, *"Risk profiling of investor is mandatory, and all investments on which investment advice is provided shall be appropriate to the risk profile of the client"* (SEBI, SEBI (Investment Advisers) Regulations 2013 [Last amended on December 08, 2016], 2016). Further, it also says that the tools need to be fit for risk profiling and the limitations should be identified and mitigated. There are further rules that require them to act in the best interests of the client (i.e. the user of the tool), disclose conflicts of interest, and store data on the investment advice given.

Under the specific rules for automated investment advisory tools, firms are required to have robust systems and controls to ensure that any advice made using the tool is suitable and in the best interest of the user. They need to disclose to the user how the tool works and the limitations of the outputs it generates. The tools mist undergo a comprehensive system audit and be subject to audit and inspection. Finally, regulations also mandate that robo advisory firms need to submit a detailed record of their process to SEBI. This includes the firm's process of risk profiling of the user and their assessment of the suitability of advice given, which is to be maintained by the investment adviser for a period of five years.

1.3. Explainable Algorithmic Decision Systems (ADS)

Algorithms are 'black-boxes' and users affected by it know little to nothing about how decisions are made. Being transparent and explaining the process helps build trust in the system and allows regulators and users to hold it accountable. With their growing ubiquity and potential impact on businesses, 'explainable AI'(xAI) or more generally, 'explainable algorithmic decision systems' is more necessary than ever.

Explainability has been defined in various ways in research. The most prominent one, given by FAT-ML considers an algorithm explainable when it can *"Ensure that* *algorithmic decisions as well as any data driving those decisions can be explained to end-users and other stakeholders in non-technical terms*". (Diakopoulos, et al.) They identify 'Explainability' as one of the five principles for accountable algorithms. The other four are responsibility, accuracy, auditability, and fairness.

The literature on explainable ADS is vast and is constantly growing. This section covers literature on the ways in which the models can be explained, the types of models that can be explained, and the trade-offs to explanations.

(Castelluccia & Le Métayer, March 2019) in their report identify three approaches to explainability. A 'black-box approach', 'white-box approach' and a 'constructive approach'. The black-box approach attempts to explain the algorithm without access to its code. In this approach, explanations are found by observing the relationship between the inputs and outputs. In the white-box approach, the code is available and can be studied to explain the decision making process. The constructive approach is a bottom-up approach that keeps explainability in mind before and while coding the ADS, thus building in 'explainability by design'.

Explainability is affected by the type of algorithm as well. While some models are easy to explain with or without access to the code, complex ML and neural network models are very difficult to explain to humans. Explainability is easier in parametric methods like linear models where feature contributions, effects, and relationships can be easily visualized and the contribution to a model's overall fit can be evaluated with variance decomposition techniques (Ciocca & Biancotti, 2018). However, that task becomes tougher with non-parametric methods like support vector machines and Gaussian processes and especially challenging in ensemble methods like random forest models. For example in fintech, an ML model used to predict loan defaults may consist of hundreds of large decision trees deployed in parallel, making it difficult to summarize how the model works intuitively (Bracke, Datta, Jung, & Sen, 2019). The newer methods of deep learning or neural networks pose the biggest challenge, they are able to model complex interactions but are almost entirely uninterpretable as it involves a complex architecture with multiple layers (Thelisson, Padh, & Celis, 2017)

(Goodman & Flaxman, 2017). Currently, there is a significant academic effort in trying to demystify these models. As it gets increasingly complex, there is also a call to avoid altogether using uninterpretable models because of their potential adverse effects for high stakes decisions, and preferably use interpretable models instead (Rudin, 2019). Several explainability methods for parametric and non-parametric models have been researched for this paper, and have been briefly covered in the methodology section.

The quality of explanations are evaluated by several indicators such as their intelligibility, accuracy, precision, completeness and consistency. There can often be a trade-off between them. By focussing on completeness, the intelligibility of the explanation can be compromised.

2. Research statement

Our research helps exlplain how an algorithm based decision making "black box", specifically in determining the risk profile of users in robo financial advisory apps, works. For this, we propose a RegTech tool to explain the algorithms decision making to regulators.

2.1. Research objective

Building user trust, especially in matters of personal wealth investment, would increase engagement with robo investment advisory services and allow users to reap the benefits they offer. Giving 'explanations' that describe the decision-making process and the parameters used for it is one way through which trust can be built. Additionally, explanations promote transparency and open it up to regulatory oversight. A regulator auditing the algorithm based on a set of pre-defined regulations or guidelines would increase user trust and ensures that automated investment advisors are unbiased, acting in the best interests of the user, and do not face a conflict of interest. With comprehensive and meaningful explanations, regulators could audit the algorithms and check if they comply with the regulations they are subject to.

Algorithms used in automated wealth/investment advisory tools are subjected to SEBI regulations in India. However, regulators without the technical knowledge possess no means to understand the algorithms and test it themselves. The objective of our research is to develop a **RegTech** tool with customized explanations that can be used by regulators to understand and evaluate the decision making of any robo-advisory application ADS.

2.2. Research Questions

1. What methods from xAI can we use to operationalize explainability in the risk-category profiling done by a robo-financial advisor algorithms?
2. Can the process of algorithmic explainability be standardised for regulators and for different data types and algorithms?
3. To what extent can these methods be used to satisfy the regulatory requirements that robo-financial advisors need to comply with?

To design a study that can explain the questionnaire based risk profiling done by robo-advisors, the boundaries of the study have to be defined. Four methodological considerations have been discussed in this section.

2.3. Defining the boundaries of the study

The first consideration for operationalizing is deciding **the depth of review/assessment** by looking at the decision-making process; this depends on the availability of required inputs for assessment. As mentioned, there is a white-box and a black-box approach. For the white-box approach, it is essential to know how the computer makes decisions. This necessitates the third party assessing the algorithm to be given access to the algorithm. While this would greatly aid transparency, they are the intellectual property and trade secrets of the robo advisory firms. This is also the case for robo-financial advisory apps. Thus, in the absence of the code, the second "black-box" approach is used. Given the "black-box" nature of algorithms, alternate methods are used to check if inputs give the intended outputs, to check the representation of training data, identify the importance of features, and find the relationship between them. Robo-financial advisors would not disclose their code or algorithm used for decision making, and hence, we will use black-box explainable methods. The firm would have to provide a dataset with a sample of its input criteria and corresponding predicted output to the regulator (i.e. the input-output data)

Second, there is a limitation to the **level of simplification** of a black box algorithm. As mentioned, there is a trade-off between complexity, completeness, and accuracy of the system and its explainability. The RegTech tool does not have access and thus does not know the algorithm used by the robo-advisor—it could be simple parametric models, the more complex non parametric models or neural networks. Our study is limited to developing a tool that can explain parametric and non-parametric models. To do this, we will employ methods from Machine Learning. Neural networks have not been tested and is not in the scope of this study.

Third, we have **global and local explanations**. Global methods aim to understand the inputs and their entire modelled relationship with the prediction target or the output. It considers concepts such as feature importance, or a more complex result, such as the pairwise feature interaction strengths (Hall & Gill, 2018). Most feature summary statistics can also be visualized by using partial dependence plots or individual conditional plots. Local explanations in the context of model interpretability try to answer questions regarding specific predictions; why was that particular prediction made? What were the influences of different features while making that specific prediction? The use of local model interpretation has gained increasing importance in domains that require a lot of trust like medicine or finances. Given the independent and complimentary value added by both methods, we will include both global and locally interpretable explanations in our study.

Finally, there is a challenge in **communicating** the results. This depends mainly on the end-user—the person who will view the explanation report. The report would have to be designed based on why they want to see the findings, and what their technical capability, statistical, and domain knowledge is. If the end-user is a layperson wanting to understand how the algorithm makes decisions at a broad level, the tool would need to be explained in a very simplified and concise manner. In contrast, if the end-user is a domain expert or a regulator who is interested in understanding the details and verifying it, the findings reported would have to reflect that detail. As mentioned in the objectives, the end user for our explanation report is a regulator. In addition to this, a branch of study called Human-Computer Interface (HCI) focuses specifically on choosing the best communication and visualization tools. Our study does not focus on this aspect, but rather confines itself to employing appropriate explainable methods for a regulator.

Hence, our tool narrows the scope of the study to the following—explaining the robo advisors black-box algorithm by approximating a best fit model to a given data set. Followed by explaining the trends and decisions observed in the dataset using global and local explanation methods. These explanations will be aimed at the regulator.

3. Methodology

The research aims to explain the questionnaire based risk profiling done by any robo-advisor using algorithms to reverse engineer key aspects of the decision making. **To study this in the absence of the algorithm, firms will have to provide regulators with the questionnaire, a sample of the user responses (input criteria) and the corresponding risk category predicted by the algorithm (i.e. the input-output data).** (part 1 of the findings quantifies the sample size that needs to be provided). Using this RegTech tool, regulators will be able to audit the algorithm to check if it complies with the regulations.

The methodology details how the tool reverse engineers the input-output data in order to understand how the algorithm takes a decision and is divided into three parts.

The first part talks about how the sample dataset required for the study was generated. In the absence of real-world data, a sample representative dataset had to be generated on which the explanation methods could be tested. To ensure that the results of the study are replicable for any type of equation used by the algorithm, we used several different methods to generate this sample dataset.

The second part looks at the information that the tool RegTech tool needs to provide to explain the robo-advisors ADS to the regulator. Information about how much each response contributes to the decision and how they relate with each other have been addressed in this section. Three explanation methods have been identified (two global and one local explanation).

In the third and final section, the technical aspects of three explanation methods for the robo advisory ADS has been detailed.

Before proceeding, we clarify the meaning of three terms that are commonly used in ML and data analysis, and explain what they mean in the context of our study (see diagram in Appendix 1)

- Each question in the questionnaire is a 'feature' in the dataset. The 'weights' associated with each feature contributes to the decision made by the ADS.
- 'Categories' refers to the options for a question (or equivalently, the response given to a question)
- The risk classes ('no risk' to 'high risk') that robo advisors assign to a user are 'classes'. There are 5 classes in this study.

That is, each question (feature) in the questionnaire has options (categories). Based on responses users can give, they are assigned a score. The output generated after answering all the questions in one out of five risk class, ranging from 'no risk' to 'high risk'.

Other definitions and terms from ML and statistics that have been used in the methodology and findings are explained in Appendix 1.

3.1. Generating the dataset for the study

To conduct this study, we needed to generate a sample data set that can adequately represent the real world. The reliability would have to be such that it can work for input-output data from any robo advisory app. In other words, the analysis should be able to handle any number of questions, any type of question (ie questions with categorical or continuous variables as its options), and any number of options. Additionally, a controlled generation of dataset allows us to build in some trends in the data. If the explanations can accurately capture these trends without access to the equations used to generate it, then we can conclude that the explanations are successful in accurately reverse engineering the decision making of the algorithm.

For our study, we surveyed several robo advisors and used the questionnaire from PayTM money to create a data set with all possible user profiles. Other robo advisory applications use similar questions therefore the choice of questionnaire is not of great importance.

Step 1- The robo advisory questionnaire is used to model an equation by giving weights to each question (i.e. feature). It is converted to a format such that output is a function of the features and weights. The equation can be represented as follows-

$$\text{output} = f(w_1x_1,\ w_2x_2,\ w_3x_3 \ldots\ w_nx_n)$$

where x_i represents the response to question 1 and w_i is any real number that represents the weight given to question1. 'f' is the function that models their relationship. For example, if the questionnaire has two questions and question 1 is about the age of the respondent and question 2 about the salary of the respondent, the output risk category could be modelled by an equation like: $\textit{risk category} = w_1(\textit{age}) + w_2(\textit{salary})$.

Step 2- A score is assigned to each option ('category') in each question. For example, within the question about age, the option with age group 18-35 could have a category score of 1, age-group 36-55 a score of 0.5 and so on. For our study, the scores assigned to each category is given in Appendix 2. The scores we have used are only indicative and have no significance. Appendix 2 explains how the features are ranked. It is important to note that the tool is valid for any input equation with any score.

Step 3- Using the questions (i.e. features) and options (i.e. categories), all possible combinations of user profiles are generated. A stratified sample of the dataset is taken for further analysis. This is equivalent to the 'input' part of the input-output dataset that the firm need to provide to the regulator.

Step 4- Using the values from step1 and step2, the output score is calculated for every user profile. The entire range of scores is divided into five risk classes in order to put each user in one of five output classes—no risk, low risk, medium risk, likes risk, and high risk. These classes are the 'output' part of the input-output dataset that the firm need to provide to the regulator.

The firm needs to provide a sample of the inputs and corresponding outputs to the regulator. The detailed process, equations used for this study and profile of the selected dataset can be found in Appendix 2.

Validity and reliability checks-

- In order to ensure that the dataset is an accurate representation of reality, data from PayTM was used. Because the process we use is independent of the number or type of features and categories, it can be replicated for any robo-advisory application.
- In order to ensure replicability, robustness and reliability of results, in step1, several types of input models were used. For our study, we tested four types of possible algorithmic equation types that could be used to generate datasets- a linear equation under independent variable assumption, an equation with interaction effects, quadratic and logarithmic generation. The details of the equations and sample is given in Appendix 2. The results for all types of equations have been reported in the findings.
- The process we use is also independent of the score associated with options (step 2). Hence, the study is valid for all values.

3.2. Information that needs to be explained by the robo-advisory

To explain the internal mechanics and technical aspects of an ADS in non-technical terms, we need to first identify the instances of decision-making which are opaque in order to make them transparent and explain them.

Robo advisors conduct a complex assessment of the users age, financial condition, risk tolerance, capacity, and more, to classify the user in to a risk class, and use it to offer automated advice suited to their investment goals. There is no way to ascertain that the advice given is not unwittingly biased, has unintended correlations or is giving undue importance to certain undesirable features (for example, the Apple credit card was accused of reproducing a gender bias because the algorithm gave a 20 times higher credit limit to a man as compared to his wife; both with the same financial background (Wired.com, 2019)). Thus, there is a need to explain the rationale for the risk classification and show that there is no undesirable importance given to certain features. In practice, if the robo advisor asks questions on age and salary, the explanation would need to tell which of the two features is more important and by how much. If gender is one of the input parameters, the explanations would be able to tell if that particular question has an undue influence on the output. Apart from this, we also need to give the regulators the ability to spot check the output. For any randomly selected user profile, a "local" explanation will allow the regulators to understand how the algorithm processes one data point and if the generated output aligns with the expected output.

In our study, we generate three explanations (two global and one local) that the regulator can use to understand how the robo-advisor takes a decision.

- **Feature importance scores**- this provides a score that indicates how useful or valuable each feature (i.e. question) is in the construction of the model. If the weightage given to a feature is large or if the feature is higher up in a decision tree algorithm, it has a higher relative importance. In our case, feature importance scores will tell us the relative importance of the features and their contribution to the risk classification.

- **Feature relations**- this tells us how features relate to each other and with the output. insights can be gained by examining the behaviour of different categories (options) within each feature (question) and how they vary with each other and affect the output. In our case, we can use these methods to find the relationships between the features, its categories and the output risk classes that are built in the black-box algorithm.

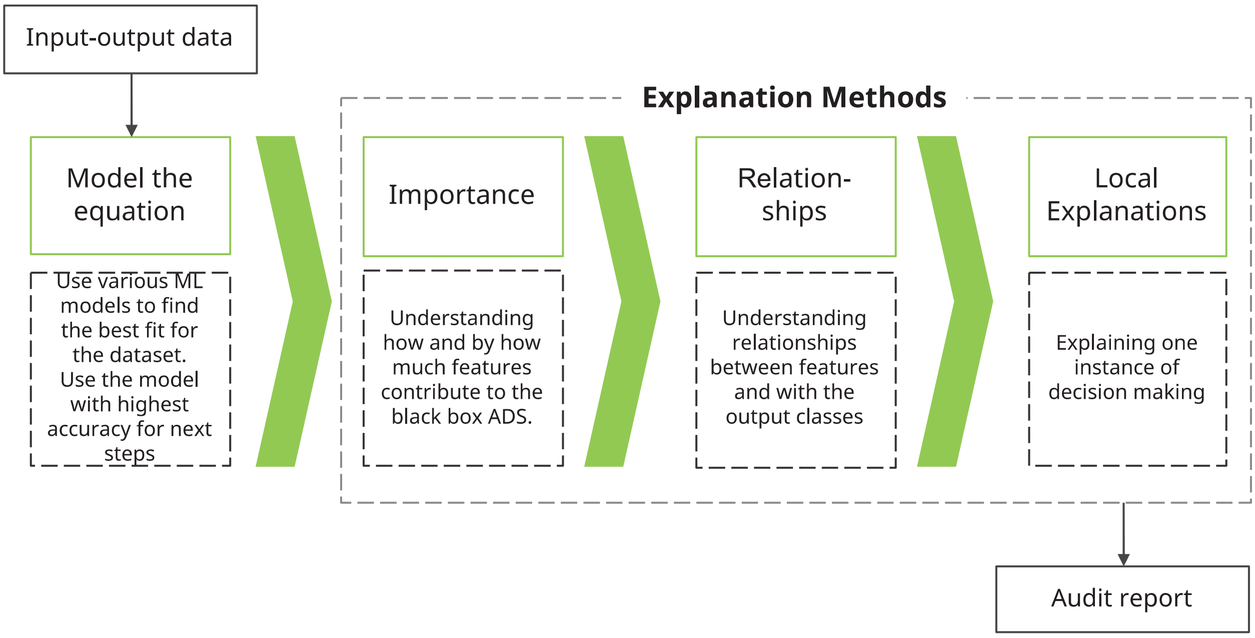

Input-output data
Model the equation
Use various ML models to find the best fit for the dataset. Use the model with highest accuracy for next steps
Explanation Methods
Importance
Understanding how and by how much features contribute to the black box ADS.
Relation-ships
Understanding relationships between features and with the output classes
Local Explanations
Explaining one instance of decision making
Audit report

Fig 1: Explanation framework. Firms would have to provide a stratified balanced sample of the input-output data to the regulator. The RegTech tool will model the data to an equation and generate an audit report with three explanations.

- **Local explanations**- Local explanations in the context of model interpretability try to answer questions regarding specific predictions; why was a particular prediction made? What were the influences of different features while making that particular prediction? As mentioned above, in our case, local explanations will help explain why a particular user was assigned a particular risk class. It can also be useful to understand boundary points and outliers.

3.3. Operationalizing the explanations

In order to find the feature importance scores, feature relations and local explanations, we reviewed and tested the several methods that researchers have developed. Various toolkits have been developed to operationalize concept of fairness, accountability, transparency and explainability in algorithms. In our review of tools, we found FairML, LIME, Aequitas, DeepLift, SHAP and VINE to be particularly relevant. Most of the toolkits focused on explainability, while only a handful try to operationalize fairness. While toolkits like FairML and LIME aim to be a generalized method or tool that is sector agnostic, others have developed techniques to address the domain-specific issues (for eg. DeepLift is used for genome sequencing). Consequently, the end product of the two approaches varies between easily understandable by all to interpretable only by domain experts. We also explored the viability of statistical methods like LASSO (least absolute shrinkage and selection operator), mRMR (minimum Redundancy Maximum Relevance) feature selection and random forest algorithms.

Firms would have to provide a stratified balanced sample of the input-output data to the regulator. The RegTech tool will model the data to an equation and generate an audit report with three explanations descrived above.

The first step is that of modelling the input-output data to an equation with no information about the method or logic used by the firm to arrive at the decision. We do this using machine learning models. Following this, the three explanations (feature importance scores, feature relations, and local explanations) are generated.

3.3.1. Modelling the dataset accurately

To model the input-output data, five **supervised ML models** are used. Firms provide the regulator a stratified sample of the input-output data. The dataset classifies inputs in five output classes (high risk to no risk) making this a multiclass type of classification. This sample is divided into two parts, the 'training data' and the 'test data'. The training data is used to train the ML models. The models then try to predict the outputs from the inputs in the test data, checks if the predicted output and the actual output match and determines the accuracy of the fit. This is repeated for multiple types of input equations to check for the reliability of the models. Overfitting is not a worry here as we are not using the model to predict new data, rather the aim of fitting a model here is to give us a better representation of the data set, and a higher accuracy indicates that the ML model is able to better reflect reality.

A variety of classifiers are available to model these mapping functions. Each ML classifier adopts a hypothesis to identify the final function that best fits the relationship between the features and output class. The input-output dataset was modelled using five machine learning algorithms frequently used for predictions; logistic regression (LR), support vector machines (SVM), decision trees (DT), naive bayes (NB) and k-nearest neighbours (KNN). These algorithms were chosen based on difference in 'hypothesis functions' and each model is good at recognizing different feature relationships and interactions. The explanation of these models and how they work can be found in Appendix 3.

The ability of the model to accurately describe the dataset is given by commonly used performance measures such as accuracy, precision, recall, and the f1 score. The definitions are given in Appendix 1. The five models are run and the model that performs best based on these metrics are selected for further explanations.

3.3.2. Finding Feature importance scores using shapley values

As mentioned, feature importance scores give the relative contributions made by each feature (question) in the risk classification decisions made by the ADS. To find these contributions we use the concept of shapley values, commonly used to decide relative contributions made by each feature in game theory. This is generated from the SHAP library, a unified framework built on top of several model interpretability algorithms such as LIME and DeepLIFT. The SHAP package can be used for multiple kinds of models like trees or deep neural networks as well as different kinds of data including tabular data and image data.

If we have 3 features (A,B,C) contributing to the output of the model then these features are permuted (B,C,A or C,A,B, etc..) to give new target values that are compared to the originals to find an error. Thus shapley values of a feature are its average marginal contributions across permutations. Shapely values are relative, thus the impacts made by each feature makes sense only in the context of other features, this means as the features/questions change we will see different patterns emerging.

3.3.3. Determining feature relations using partial dependence plots

Once the important features are identified, we need to assess the interactions and relationship between them (or a subset) and the response. This can be done in many ways, but in machine learning it is often accomplished by constructing partial dependence plots (PDPs), and we use this method in our study. These plots portray the marginal effect one or two features have on the output risk classes and visualizes the relationship.

PDP can be used as a **model agnostic global level understanding** method to gather insights into black box models. Model agnostic means that PDP's make no assumptions regarding the underlying model. The partial dependence function for regression is defined as-

$$f_{x_s}(x_s) = E_{x_c}[f(x_s, x_c)] = \int f(x_s, x_c)dP(x_c)$$

x_s is the set of features we find interesting, x_c is the complement of that set (set of all features we don't find interesting but are present in the dataset), $f(x_s)$ gives the partial dependence and $P(x_c)$ is the marginal probability density of x_c. f is the prediction function. The whole function $f(x_s)$ is estimated as we don't know the f (it's model agnostic) nor do we know the marginal probability distribution.

$$f_s = \frac{1}{N}\sum_{i=1}^{N} f(x_s, x_{c_i})$$

The approximation here is twofold: we estimate the true model with f, the output of a statistical learning algorithm, and we estimate the integral over x_c by averaging over the Nx_c values observed in the training set.

3.3.4. Local explanations

Local explanations mean explaining a single instance of decision made by an ADS system. To find the logic behind these decisions we used LIME, or Locally Interpretable Model agnostic Explanations. This method, developed by a group of researchers, uses local surrogate models to approximate the predictions of the underlying black-box model. Local surrogate models are interpretable models like Linear Regression or a Decision Trees that are used to explain individual predictions of a black-box model (Ribeiro, Singh, & Guestrin, 2016). LIME trains a surrogate model by generating a new data-set out of the datapoint of interest. The way it generates the data-set varies dependent on the type of data. For text and image data LIME generates the dataset by randomly turning single words or pixels on or off. In the case of tabular data, LIME creates new samples by permuting each feature individually. The model learned by LIME generally is a good local approximation of the black box model and gave satisfactory results for our study.

4. Findings

The findings are divided in four parts. The first part gives the results of the Machine Learning models that are used to fit the input-output data and reverse engineer the importance of features in the robo advisors risk profiling. The second and third part explains the risk profiling using global explanation methods. The second part reports the feature importance scores and the third part reports the feature relations. The fourth and final part of the findings provides the local explanations to spot-check the algorithm or explain one specific decision made by it.

The findings have been reported for select cases. All test cases, generation of sample data, and findings can be accessed through this github link- https://github.com/NehaSontakk/Algorithmic-Explainability-in-Risk-Profiling-done-by-Robo-Advisors

Part 1- modelling the risk profiling decision

The aim of the first part is to fit a model to the input-output data that can predict the outputs as accurately as possible. As mentioned in the methodology, in this step, various ML models are used and the most accurate model is identified. This first step is crucial because the best-fit model is required to implement the three explanation methods.

The accuracy of the prediction and the f1-scores of the classes need to be considered together to select the best model for the dataset. The results (for five ML models and four input equations) have been summarized in the table below (Table 1).

Table 1 accuracy of the prediction and the f1-scores of the classes for five models (LR, GNB, KNN, SVM, DT), and four input equations (linear equations under independent variable assumption, quadratic, equations with interaction effects and logarithmic)

	Linear Equation under independent variable assumption		**Quadratic Equation**		**Equations with interaction effects**		**Logarithmic Equation**	
Performance Metrics	**Accuracy (%)**	**F1 - Score**	**Accuracy (%)**	**F1 - Score**	**Accuracy (%)**	**F1 - Score**	**Accuracy (%)**	**F1 - Score**
Logistic Regression (LR)	90	• no risk : 0.49 • low risk : 0.91 • moderate : 0.93 • likes risk : 0.86 • high risk : 0.52	78	• no risk : 0.88 • low risk : 0.77 • moderate : 0.73 • likes risk : 0.71 • high risk : 0.91	78	• no risk : 0.96 • low risk : 0.79 • moderate : 0.23 • likes risk : 0.80 • high risk : 0.94	76	• no risk : 0.91 • low risk : 0.77 • moderate : 0.00 • likes risk : 0.78 • high risk : 0.94
Gaussian Naive Bayes (GNB)	75	• no risk : 0.56 • low risk : 0.71 • moderate : 0.81 • likes risk : 0.65 • high risk : 0.26	70	• no risk : 0.79 • low risk : 0.76 • moderate : 0.68 • likes risk : 0.51 • high risk : 0.72	67	• no risk : 0.95 • low risk : 0.82 • moderate : 0.43 • likes risk : 0.00 • high risk : 0.40	68	• no risk : 0.77 • low risk : 0.61 • moderate : 0.58 • likes risk : 0.64 • high risk : 0.80
K - Nearest Neighbours (KNN)	93	• no risk : 0.90 • low risk : 0.94 • moderate : 0.94 • likes risk : 0.92 • high risk : 0.86	96	• no risk : 0.97 • low risk : 0.96 • moderate : 0.96 • likes risk : 0.95 • high risk : 0.96	97	• no risk : 0.99 • low risk : 0.98 • moderate : 0.96 • likes risk : 0.95 • high risk : 0.94	98	• no risk : 0.98 • low risk : 0.97 • moderate : 0.96 • likes risk : 0.98 • high risk : 0.99
Support Vector Machines (SVM)	98	• no risk : 0.63 • low risk : 0.97 • moderate : 0.99 • likes risk : 0.99 • high risk : 0.94	93	• no risk : 0.94 • low risk : 0.92 • moderate : 0.93 • likes risk : 0.93 • high risk : 0.94	96	• no risk : 0.96 • low risk : 0.95 • moderate : 0.95 • likes risk : 0.95 • high risk : 0.95	92	• no risk : 0.92 • low risk : 0.89 • moderate : 0.87 • likes risk : 0.94 • high risk : 0.96
Decision Trees (DT)	89	• no risk : 0.81 • low risk : 0.89 • moderate : 0.90 • likes risk : 0.88 • high risk : 0.81	96	• no risk : 0.97 • low risk : 0.95 • moderate : 0.95 • likes risk : 0.95 • high risk : 0.95	98	• no risk : 0.99 • low risk : 0.98 • moderate : 0.97 • likes risk : 0.95 • high risk : 0.94	99	• no risk : 0.99 • low risk : 0.99 • moderate : 0.99 • likes risk : 0.99 • high risk : 0.99
Best Model	K - Nearest Neighbours		K - Nearest Neighbours		Decision Tree		Decision Tree	
Explanation	K - Nearest Neighbours can generate a highly convoluted decision boundary, hence points that are very close to each other can be modelled very well using this method				DTs perform very well for all input equations except the linear model. It gives very accurate results because the options are categorical, which DT can identify much better			

As our findings show, KNN fits linear (under independent variable assumption) and quadratic equations most accurately and the Decision Tree model fits equations with interaction effect and logarithmic equations most accurately. The findings also highlight why it is not sufficient to consider only the accuracy. Take the example of SVM on a linear equation. It gives a high accuracy of 98%, higher than the KNN model. However, the f1 score of the no-risk class is only 0.63. This indicates that the SVM model can make very good predictions for other classes, but fails to do it in the no-risk class.

The RegTech tool will run the sample input-output data provided by the firm. The four ML models will model it. The model that maximizes accuracy and f1-score will be selected and used as a basis for generating the explanations.

Optimal size of input-output sample data that the RegTech tool requires

What is the minimum size of training data that firms should share with the regulator without compromising the accuracy of the modelling? While there are thumb rules and more is considered better, we report the minimum sample required. To find this,

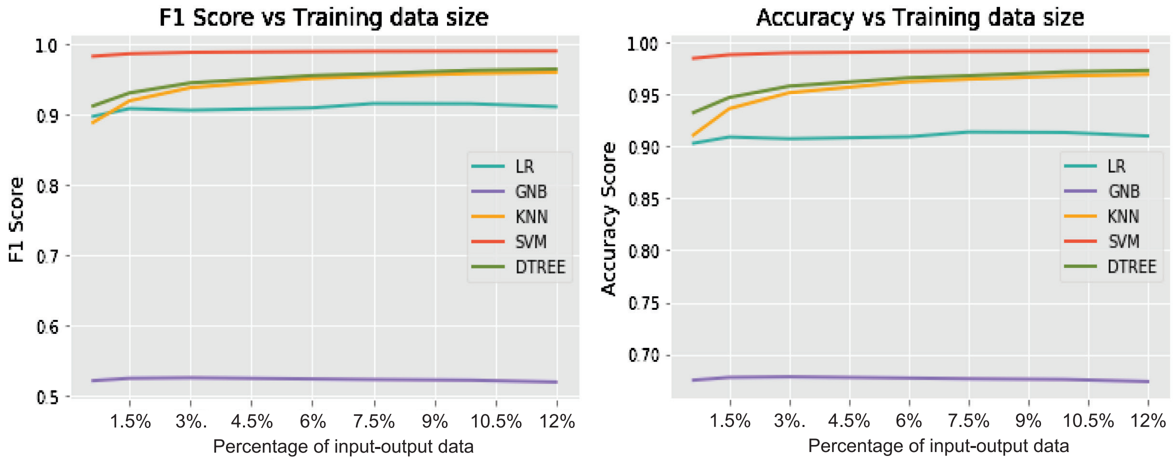


Fig 2: Graph on left- F1 scores (y-axis) versus percentage of input-output data used as training data (y-axis). Graph on right- Accuracy (y-axis) versus percentage of input-output data used as training data (y-axis).

Lines of different colours represent the results for different ML models.

LR- linear regression; GNB- Gaussian Naïve Bayes; KNN- K-Nearest Neighbours; SVM-Support Vector Machines; DTREE- Decision Tree we ran the models with different sample sizes in order to provide a ball-park figure or the number of data points that need to be provided by the robo-advisory firm to the regulator.

Stratified samples of the input-output data of different sizes were selected as the training data, the ML models were run on them and the accuracy and f1 scores were found. The sample sizes included values between 1.5% of the training data to 12% of the training data As expected, the accuracy the directly proportional to the sample size— considering a larger sample gives greater accuracy. However, findings show that relationship is not linear. The accuracy of most models increases with the increase in sample size till about 6% of the data and then stabilizes. Amongst all the models, SVM (Support vector machine) performs the best with all sizes of data, followed by KNN and the DT model.

A 6% stratified sample of all input-output data, which translates to 67500 data points, would be sufficient in our case to run these models. Therefore, firms would need to give the regulators a minimum of 6% of the training data or ~67,500 data points, whichever is higher.

Part 2- Feature importance scores

Feature importance scores are part of the global explanations and have been found using the SHAP values. They have been represented using SHAP plots. They tell how and by how much each feature (question) contributes to the ADS risk classification process. We report two importance scores- the feature importance and the class-wise feature importance.

Importance scores for all equation types used to generate the dataset for the study were calculated. However, in the following sub-sections of the findings, the results showcase the scores obtained for equations that have interaction effects.

It is important to remember that these explanation methods are replicable for any set on input features, including demographic features (like gender, race), behaviour (such as purchase history or internet activity) or opinions (like political leaning).

4.2.1. Feature importance-

Figure 3 shows that 'Age' is the most important feature in the model, and has the greatest contribution to the risk categorization process. This accurately represents the weights that were given when the dataset for the study was generated, indicating that the explanation method is successful in reverse engineering the input-output data without having access to the model. This will allow regulators to understand if any undesirable feature has a disproportionate importance score.

4.2.2. Class-wise feature importance

Feature importance scores help understand the importance of questions. Class-wise feature importance plots show how the categories in each feature behaves differently in different output class ('no risk' to 'high risk') and quantifies the effect. For instance, if a person has a large loan amount to repay every month, their response should negatively contribute to the high risk class and positively contribute to the low risk class. Further, it shows the relative importance of features and the distribution of the stratified sample in the output class.

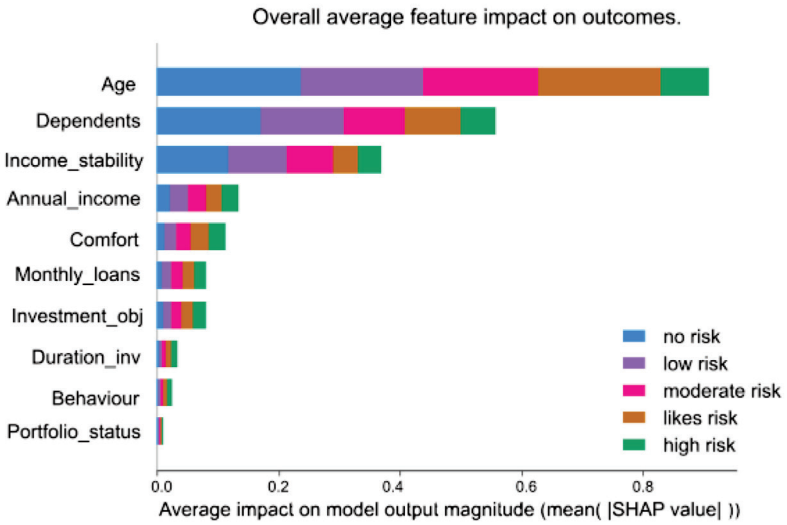
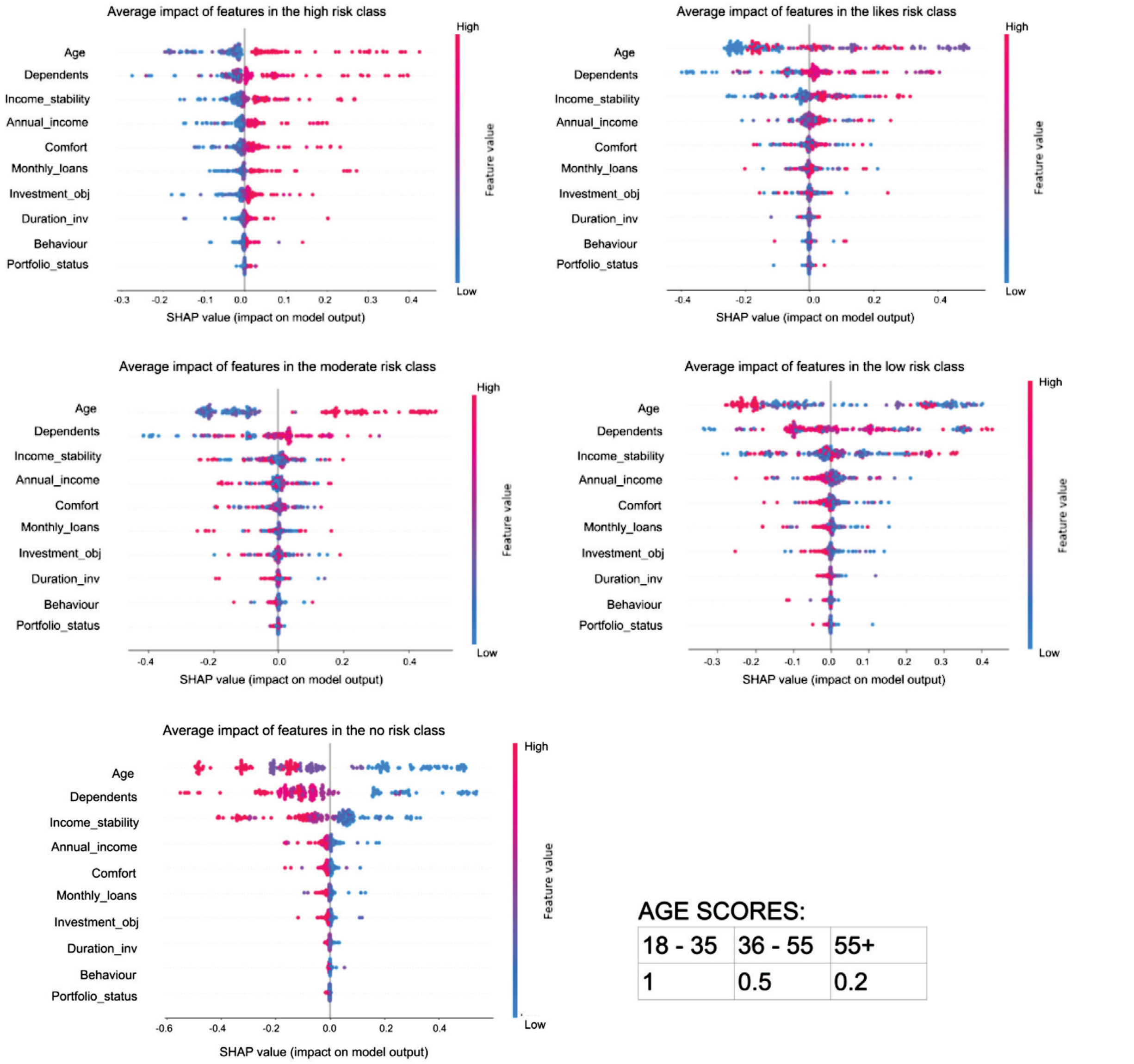


Fig. 3: Average impact of features on risk classes. The y-axis lists the features used to decide the risk class in descending order of importance. The x-axis shows the shapely value that quantifies the influence. The length of the bar indicates the total contribution of the feature to the output class. The colours indicate the average contribution of the feature different risk classes.

The SHAP plot shows that AGE is the most important feature while predicting the risk class for all output classes ('no risk' to 'high risk')



AGE SCORES:

18 - 35	36 - 55	55+
1	0.5	0.2

Fig 4: Class-wise feature importance plots for the five output risk classes ('no risk' to 'high risk'). The features are listed on the y-axis (in descending order of feature importance) and shapely values that quantify the effect are shown on the x axis. The colour represents the score of the categories in a feature. The distribution of points in each feature represents the distribution of data points in the sample and also shows the extent of negative or positive influence.

The scores for the feature 'age of the user' that is used by the ADS is shown in the table. A person whose age falls in the 18-35 category is assigned a score of 1.

Using the 'age' feature as an example to interpret the graphs, it can be seen that a young person (in the age category of 18-35) has a high category score (of 1). Thus, according to the first graph in Figure 4, this demographic feature would result in a large positive contribution to the 'high risk' output (a positive shapely value of ~0.4). Similarly, an older person (age category of 55+ and a small category score of 0.2) will negatively contribute to the ‘high risk’ class. Additionally, features like the investment objective and monthly loans contribute very less to the extreme classes (‘high risk’ and ‘no risk’), but influence the output significantly in the ‘moderate risk’ class. Once again, we observe that these plots can accurately represent the trends in the model without having access to it.

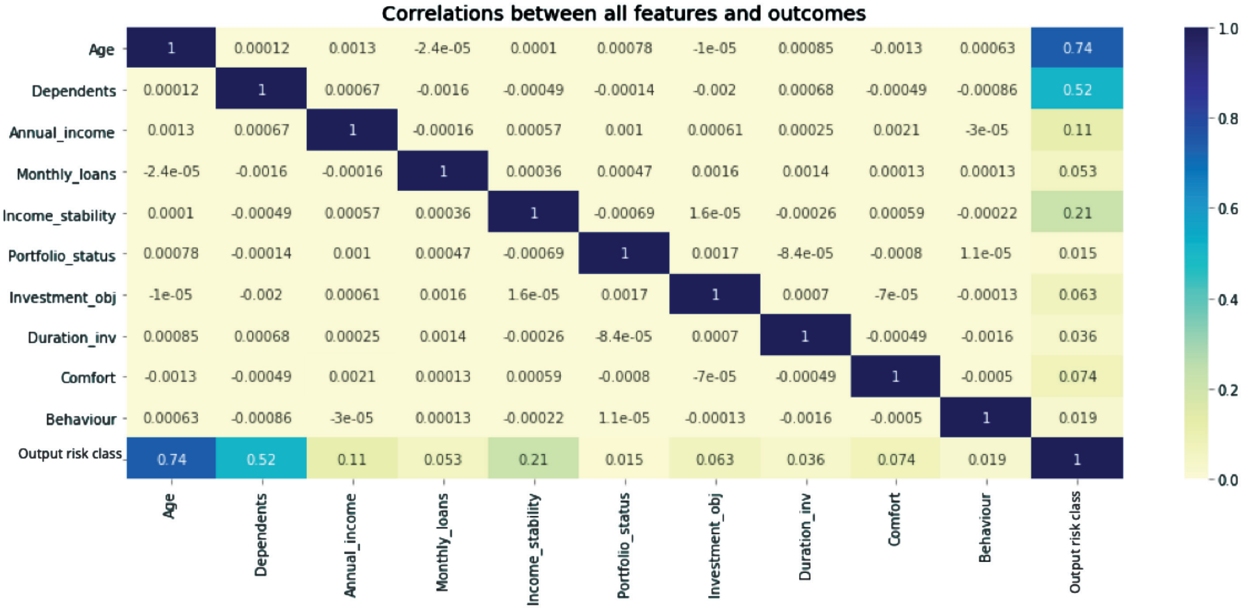
Hence, using this, regulators can understand the how various categories in a feature (for example gender being female) an affect an output, and by how much.

Part 3- Relationships

This section reports the relationship between features and the output class by showing how the changes in one or more feature changes the output.

One simple way of finding the relationships is to see the correlations between features and between features and the output, as shown in the correlation matrix in Fig 5.

Relationships can also be visualized using partial dependence plots between one feature and the output or two features and the output. Fig 6 shows the partial dependence relationships between one feature (age) and the output risk class decision.



Correlations between all features and outcomes

	Age	Dependents	Annual_income	Monthly_loans	Income_stability	Portfolio_status	Investment_obj	Duration_inv	Comfort	Behaviour	Output risk class
Age	1	0.00012	0.0013	-2.4e-05	0.0001	0.00078	-1e-05	0.00085	-0.0013	0.00063	0.74
Dependents	0.00012	1	0.00067	-0.0016	-0.00049	-0.00014	-0.002	0.00068	-0.00049	-0.00086	0.52
Annual_income	0.0013	0.00067	1	-0.00016	0.00057	0.001	0.00061	0.00025	0.0021	-3e-05	0.11
Monthly_loans	-2.4e-05	-0.0016	-0.00016	1	0.00036	0.00047	0.0016	0.0014	0.00013	0.00013	0.053
Income_stability	0.0001	-0.00049	0.00057	0.00036	1	-0.00069	1.6e-05	-0.00026	0.00059	-0.00022	0.21
Portfolio_status	0.00078	-0.00014	0.001	0.00047	-0.00069	1	0.0017	-8.4e-05	-0.0008	1.1e-05	0.015
Investment_obj	-1e-05	-0.002	0.00061	0.0016	1.6e-05	0.0017	1	0.0007	-7e-05	-0.00013	0.063
Duration_inv	0.00085	0.00068	0.00025	0.0014	-0.00026	-8.4e-05	0.0007	1	-0.00049	-0.0016	0.036
Comfort	-0.0013	-0.00049	0.0021	0.00013	0.00059	-0.0008	-7e-05	-0.00049	1	-0.0005	0.074
Behaviour	0.00063	-0.00086	-3e-05	0.00013	-0.00022	1.1e-05	-0.00013	-0.0016	-0.0005	1	0.019
Output risk class	0.74	0.52	0.11	0.053	0.21	0.015	0.063	0.036	0.074	0.019	1

Fig. 5: The correlation matrix shows the correlation coefficient between features and with the output class. A dark colour indicates a higher correlation.

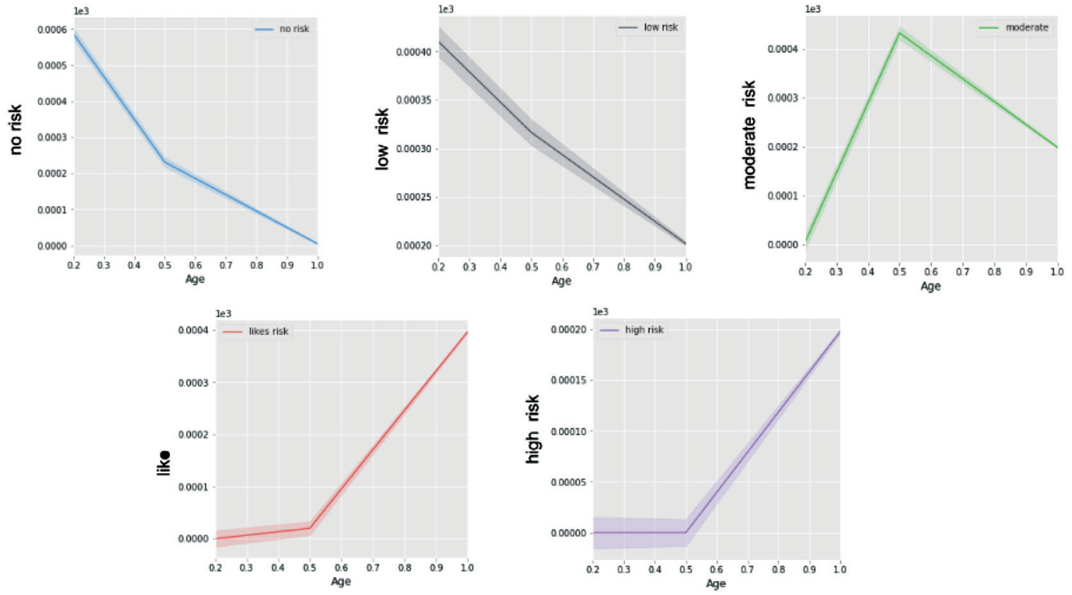


Fig 6: Relationship of AGE with output risk classes ('high risk' to 'no risk'). As age increases (i.e. the age score decreases), the contribution to 'high risk' class decreases. For the moderate risk class, there is an inflection point indicating that that very high or very low age scores would negatively contribute to the 'moderate risk' class.

This shows how the changing scores of categories in a feature relate with the output class. It helps visualize the shape and the infliction point of the relationship, allowing the regulator to identify breaks where the effect of a feature could change drastically.

Similar partial dependence plots can be drawn to identify the relationship between two features and the output.

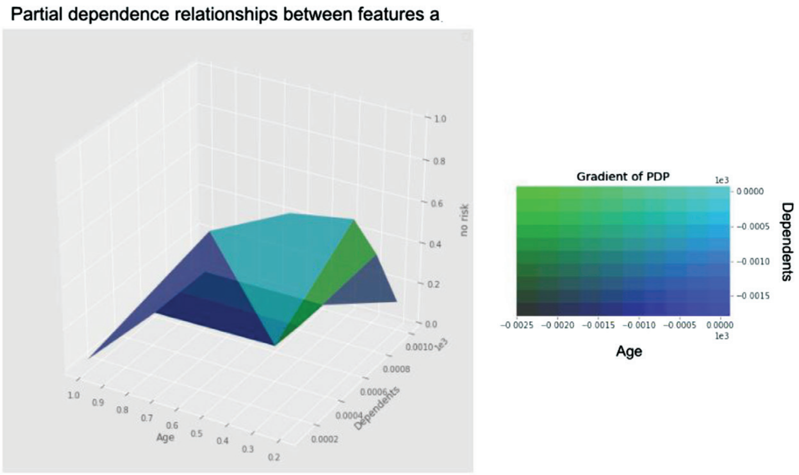


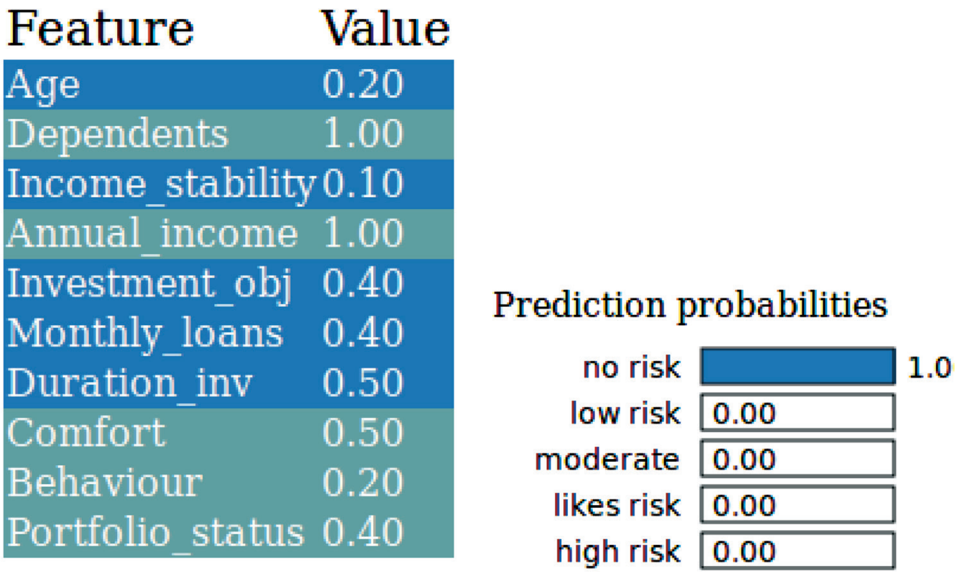
Fig. 7: Relationship between AGE, DEPENDENTS and the 'low-risk' class output. The 3D graph on the left shows the features in the axes of the horizontal plane and the output risk class ('low risk') in the vertical axis. The graph on the right explains the colour gradient seen in the PDP. The plot shows how the low-risk taking output changes with different combinations of 'age' and 'dependence'.

This shows how the movement of two features influence the output. In the RegTech tool, the various features whose relationship the regulator wants to observe can be selected and the plots can be created dynamically.

Part 4- Local explanations

'Local' explanations using LIME explain the features that influence a single observation. In the explanation reports, the regulators can randomly select an input condition to understand how the features in that condition affects the output risk class. The report would give the weights of the features influencing the predicted output class (Fig 8) and the influence of the input features on all possible output classes (Fig 9).

In Fig 8, the contributions of each feature to and against every class are shown. The highest contributions made by the top features are in the 'no risk' class, all other class contributions are negligible thus the final prediction is 'no risk'.



Feature	Value
Age	0.20
Dependents	1.00
Income_stability	0.10
Annual_income	1.00
Investment_obj	0.40
Monthly_loans	0.40
Duration_inv	0.50
Comfort	0.50
Behaviour	0.20
Portfolio_status	0.40

Fig. 8: For one randomly selected input condition, the table on the left shows the feature values of the input condition and the colour shows the influence it has on a the different output classes (shown in the right). It shows that the Age, income stability, investment objectives, monthly loans and duration of investment (features in blue) of the user are the primary determining factors that classify the user to the 'no risk' class with a high probability. The features in green (number of dependents, annual income, comfort, behaviour and portfolio status) push the classification towards another output class, however, its effect is negligible.

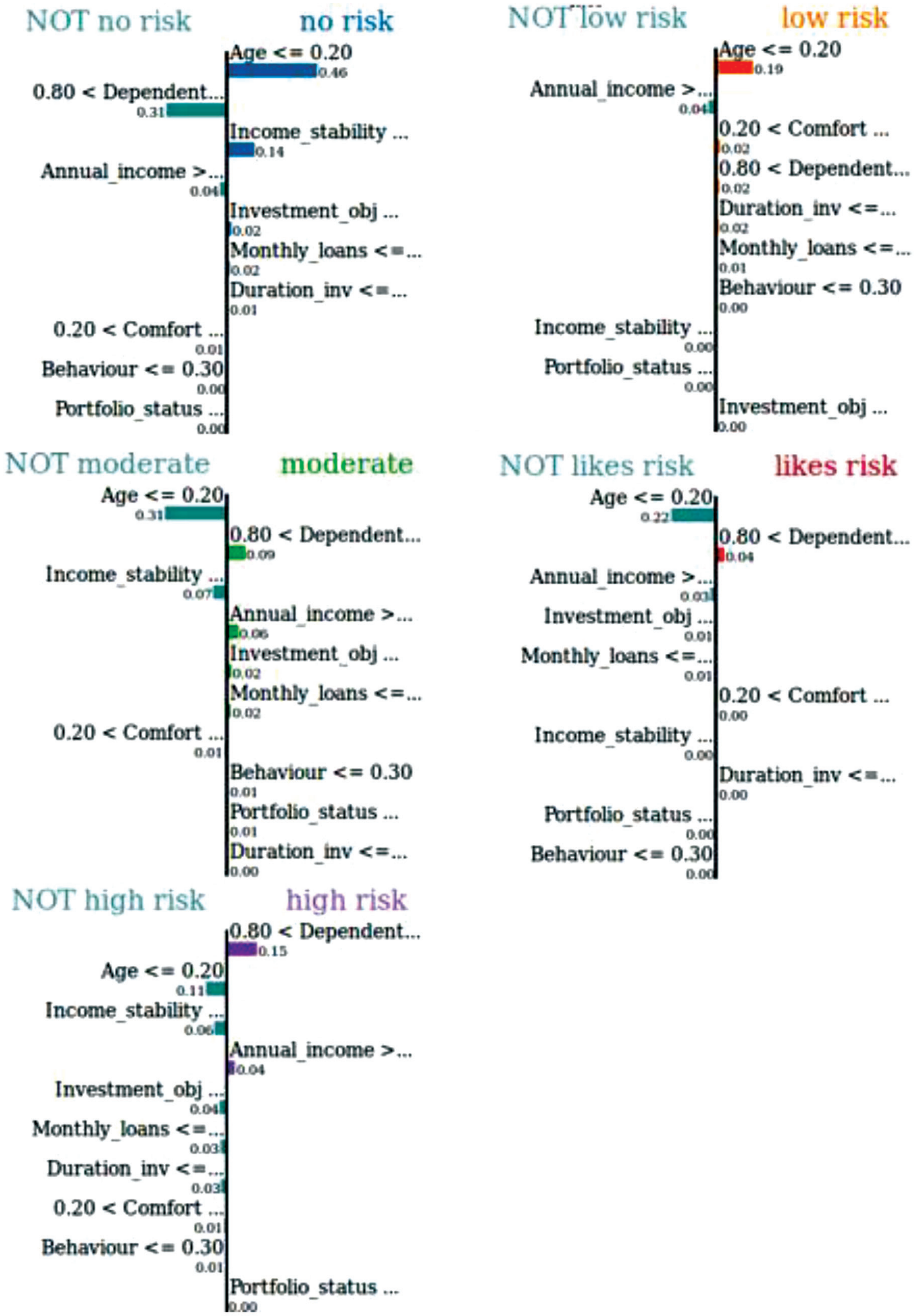


Fig. 9: Class probability for each feature in the observation. For each output class ('no risk' to 'high risk'), the each graph show how the features contribute to the probability of either falling in the class (on the positive axis) or NOT falling in the class (negative axis). In the 'no risk' class, the Age feature strongly pushes it towards 'no risk'.

Fig 9 shows how the features in same input condition contribute to the different output classes. In this case, the overall sum of probability lies in the 'no risk' class. The Age feature that matters most has a high probability of belonging to the no risk class.

Using this, regulators can understand a single random observation and understand how the algorithm classifies it to the out class, and hence spot check the algorithms decision making.

5. Conclusion

In this study, we achieve the following— (i) operationalizing explainability in the case of robo-advisory risk profiling by creating a RegTech tool that can be used for several algorithms and use cases (ii) describing how this could be used by fintech regulators to audit algorithms and check if they comply with the regulations that they are subject to.

We do this for black-box algorithms where firms have to provide a stratified balanced sample of the input-output data, and the regulator uses the RegTech tool to model the data to an equation and generate an audit report with three explanations (consisting of two global and one local explanation method). With this, regulators can understand how each question contributes to the output, how they relate to each other and conduct spot checks. We find that the methods used are able to model the dataset with high degree of accuracy and provide accurate explanations. The methods have been tested using various input conditions to ensure its reliability.

Revisiting the SEBI rules for automated tools in investment advisory, our study has proposed an approach to check if the automated tools comply with the regulations. Using the RegTech tool, we can subjecting the tool to a comprehensive system audit and inspection. Further, we can provide an explanation for how the tools algorithm works. While an explanation for the algorithm is not mandated, the regulator can use this to check if the robo-advisory tool acts in the best interest of the client without any unintended machine bias.

It is important to note that these explanation methods are replicable for any set on input features, including demographic features (like gender, race), behaviour (such as purchase history or internet activity) or opinions (like political leaning).

Thus, our approach has the potential to enhance the technical capabilities of capital markets regulator without the need for in-house computer science expertise. Considerable work and research would be required to create a comprehensive tool capable of operationalizing all regulations.

6. Discussion and way forward

With algorithms permeating various aspects of public life, they are increasingly being subject to scrutiny and regulations. However, designing and implementing regulations without knowledge of how an algorithmic system works and what its externalities are would prove to be ineffective. To formulate regulations that work, they need to be informed by the technical and operational limitations while also considering the ethical aspects. This is especially true for the case of ADS, where there are glaring problems and yet there is a struggle to enforce concepts like fairness, accountability and transparency. As (Goodman & Flaxman, 2017) point out, the GDPR acknowledges that the few if any decisions made by algorithms are purely ‘'technical”, and the ethical issues posed by them require rare coordination between ‘technical and philosophical resources’. Hence, we need dialogue between technologist and regulators and they need to work together to design safeguards by pooling their domain knowledge.

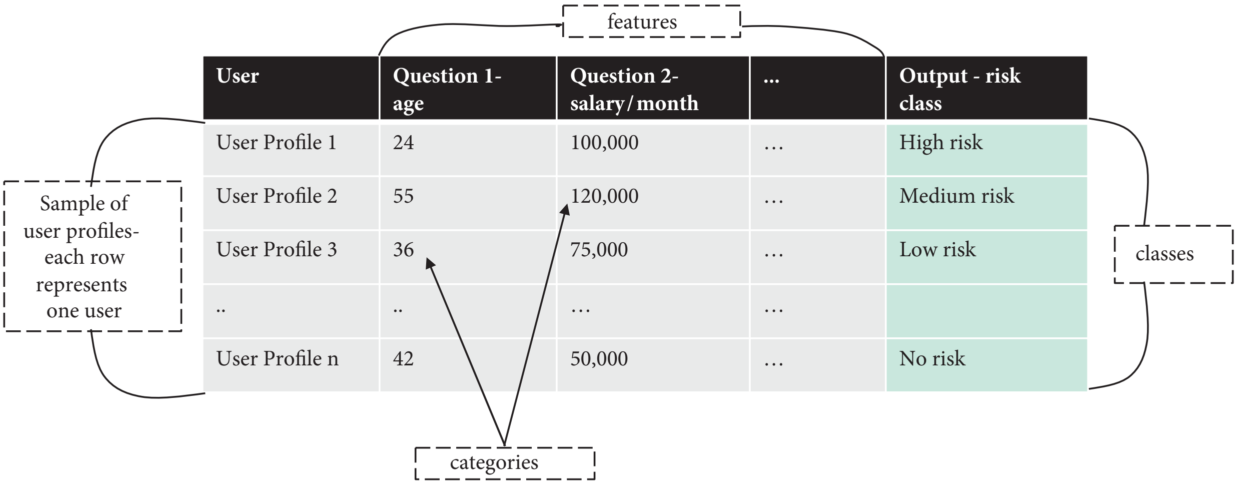
One way to achieve this is by creating regulatory sandboxes. Sandboxes act as test beds where experiments can happen in a controlled environment. They are initiated by regulators for live testing innovations of private firms in an environment that is under the regulators supervision (Jenik & Lauer, 2017). It can provide a space for dialogue and developing regulatory frameworks for the speed at which technological innovation happens, in a way that “doesn’t smother the fintech sector with rules, but also doesn’t diminish consumer protection" (BBVA, 2017). This method would help build collaborative regulations and also open up the dialogue of building in explainability by design in ADS early on in the process.

Future work needs to be on the regulatory and technical front. On the regulatory front, we need to work with the regulators to understand the grasp-ability of various explanation methods. Appropriate explanations also need to be extended to the user.

On the technical front, our work can be expanded to include increasingly more complex situations. A standardized and robust documentation process for algorithms also needs to be initiated to maintain accountability and makes it easier to audit the system.

Appendix

Appendix 1- Definitions and key terms



User	Question 1- age	Question 2- salary/month	...	Output - risk class
User Profile 1	24	100,000	…	High risk
User Profile 2	55	120,000	…	Medium risk
User Profile 3	36	75,000	…	Low risk
..	..	…	…	
User Profile n	42	50,000	…	No risk

1. Feature- A feature is a measurable property of the object we are trying to analyze. In datasets, features appear as columns[1].
2. Accuracy- Accuracy gives the percentage of correctly predicted samples out of all the available samples.

[1] https://www.datarobot.com/wiki/feature/

Accuracy is not always the right metric to consider in imbalanced class problems; in the risk dataset, class 2 has the most samples greatly outnumbering samples in class 1 and 5. This could mean that even if most samples are incorrectly labelled as belonging to class 2 then the accuracy would still be relatively high giving us an incorrect understanding of the models working. Just considering the accuracy, the most accurate classifier is the decision tree, closely followed by knn and svm who supersede the logistic regression and naive bayes classifiers.

3. Recall- the ability of a model to find all the relevant samples. This gives the number of true positive samples by the sum of true positive and false negative samples. True positive samples are the samples correctly predicted as true by the model and false negatives are data points the model identifies as negative that actually are positive (for example points that belong to class 2 that are predicted as not belonging to class 2).

   For example, in the performance metrics for logistic regression we find that the performance is thrown off by class moderate/medium -risk takers, this is most probably because the class has too many samples in the training data causing it to overfit(logistic regression is prone to overfitting).

4. Precision- it is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative, or in our example, individuals the model classifies as class 2 that are not. While recall expresses the ability to find all relevant instances in a dataset, precision expresses the proportion of the data points our model says was relevant actually were relevant.

5. F1 score- Sometimes trying to increase precision can decrease recall and vice versa, an optimal way to combine precision and recall into one metric is by using their harmonic mean also called the F1-Score.
   F1 = 2* (precision*recall)/(precision + recall)

Appendix 2- Details of sample dataset generation that has been used for this study

We generated a dataset by permuting all possible sequences of the answers for each question (i.e. categories for each feature) asked by prominent robo advisory apps in India. In this case, we used the questions from PayTM money. The flow graph below visualizes the importance of the features and the most important variables. table shows the frequently asked questions in robo-advisory apps with corresponding options.

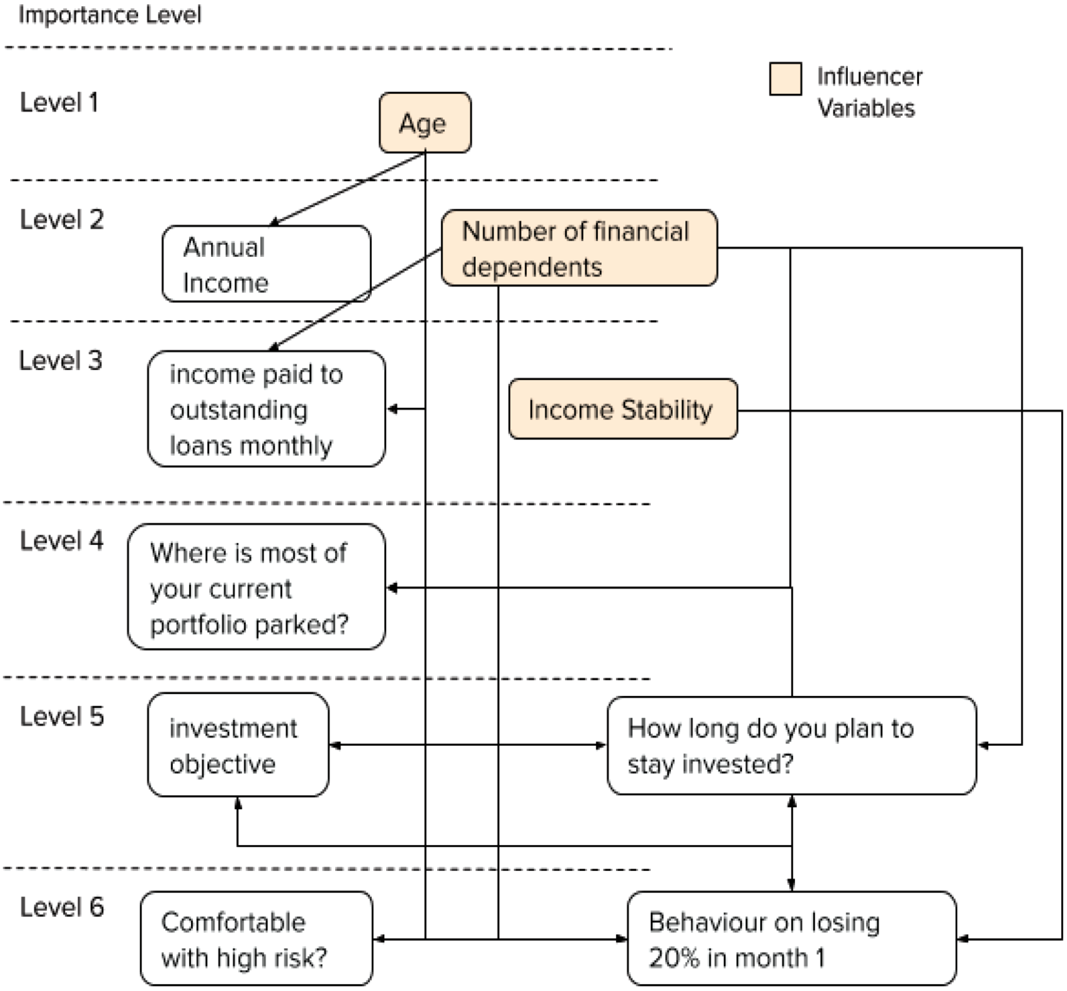
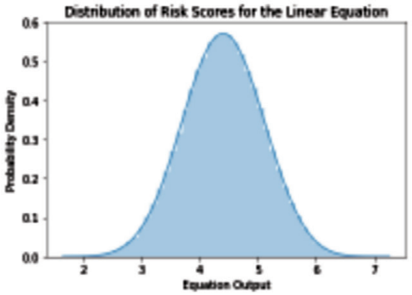
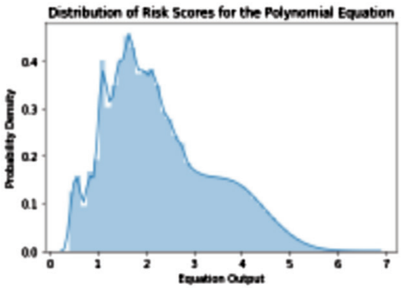
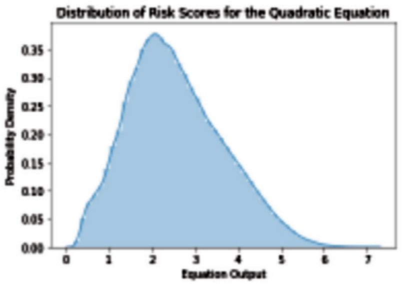
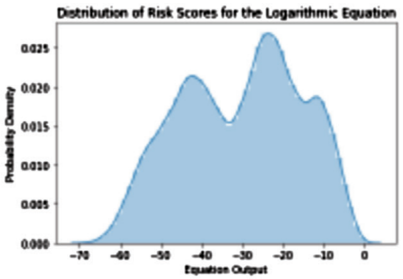
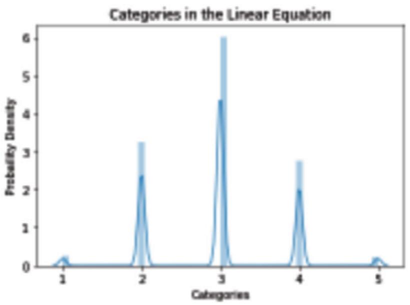
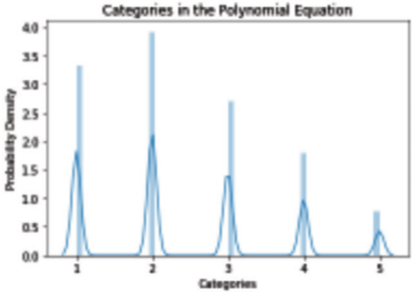
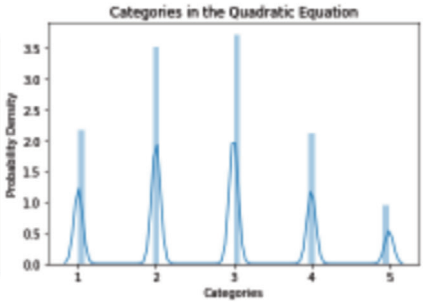
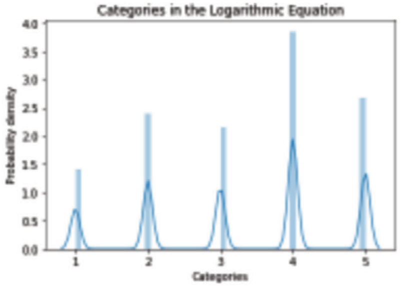

Importance Level
Influencer Variables
Level 1
Age
Level 2
Annual Income
Number of financial dependents
Level 3
income paid to outstanding loans monthly
Income Stability
Level 4
Where is most of your current portfolio parked?
Level 5
investment objective
How long do you plan to stay invested?
Level 6
Comfortable with high risk?
Behaviour on losing 20% in month 1

Table 2 Frequently asked questions in robo-advisory apps with corresponding options. The weights to the questions (features) and scores given to the options (categories) were set at our discretion in order to generate the dataset. The values are for representation and the method would work for any set values.

Variable names	Questions	Weight	Option1	Score (option 1)	Option2	Score (option 2)	Option3	Score (option 3)	Option4	Score (option 4)	Option5	Score (option 5)
x(1,1)	What's your age (in years)	1	18-35	1	36-55	0.5	55+	0.2				
x(2,1)	How many people depend on you financially?	0.83	No one	1	Spouse only	0.8	spouse and children	0.6	Parents only	0.6	Spouse, children and parents	0.1
x(2,2)	What's your annual income range?	0.83	Below INR 1 lac	0.2	Between INR 1 Lac - INR 5 Lac	0.4	Between INR 5 lac - 10 Lac	0.6	Between INR 10 Lac - INR 25 Lac	0.8	Above 25 Lac	1
x(3,1)	What % of your monthly income do you pay in outstanding loans, EMI etc?	0.65	None	1	Up to 20% of income	0.8	20-30% income	0.6	30-40% of income	0.4	50% or above of income	0.2
x(3,2)	Please select the stability of your income	0.65	Very low stability	0.1	Low stability	0.3	Moderate Stability	0.6	High Stability	1	Very high stability	1
x(4,1)	Where is most of your current portfolio parked?	0.5	Savings and fixed deposits	0.4	Bonds/debt	0.6	Mutual Funds	0.5	Real Estate or Gold	0.4	Stock Market	0.8
x(5,1)	What's your primary investment objective?	0.8	retirement planning	0.65	Monthly Income	0.6	Tax Saving	0.4	Capital Preservation	0.5	Wealth Creation	1
x(5,2)	How long do you plan to stay invested?	0.8	Less than 1 year	0.5	1 to 3 years	0.8	3 to 5 years	0.65	5 to 10 years	0.6	more than 10 years	0.7
x(6,1)	To achieve high returns, you are comfortable with high risk investments	0.7	Strongly agree	1	Agree	0.9	Neutral	0.5	Disagree	0.2	Strongly disagree	0.1
x(6,2)	If you lose 20% of your invested value one month after investment, you will	0.65	Sell and preserve cash	0.2	Sell and move cash to fixed deposits or liquid fund	0.3	Wait till market recovers and then sell	0.5	Keep investments as they are	0.8	Invest more	1

Table 3 Details of the sample input-output data generated for the study using four equations.

Types of Equations	Linear Equation under Independent Variable assumption	Equation with interaction effects	Quadratic Equation	Logarithmic Equation
Equations	w_11*Age + w_21*Dependents + w_22*Annual_Income + w_31*Monthly_loans + w_32*Income_stability + w_41*Portfolio_status + w_51*Investment_obj + w_52*Duration_inv + w_61*Comfort + w_62*Behaviour = output	w_11*Age + w_21*Dependents + w_22*[k]*Age + w_31*x_31[l]*Age*Dependents + w_32*x_32[m] + w_41*x _41[n]*Age*Dependents *x_52[o] + w_51*x_51[o]*Age*Dependents *x_52[p] + w_52*x_52[p]*x_51[o]* Age*Dependents + w_61*x_61[q]*Age*Dependents + w_62*x_62[r]*Age* Dependents*x_32[m]*x_52[p]	w_11(*Age*** *3)* + *w_21* **Age* (Dependents* *2)* + *w_22* **Age* Annual_Income + w_31(*Monthly_loans*** *2)* + *w_32*(Income_stability* *3)* + *w_41* Dependents* Portfolio_status + w_51(*Investment_obj*** *2)* Monthly_loans + w_52* Duration_inv* Dependents + w_61* Monthly_loans* Comfort + w_62* Behaviour* Dependents	w_11*3*math.log(Age,3) + w_21*2*math.log(Age*Dependents, 2) + w_22*3*math.log(Age*Annual_Income,2) + w_31*3*math.log(Age*Monthly_loans,2) + w_32*3*math.log(Age*Income_stability,2) + w_41* Portfolio_status + w_51*3*math.log(Age* Income_stability*Investment_obj,2) + w_52*Duration_inv *Behaviour + w_61*2*math.log(Comfort*Age,2) + w_62*Behaviour* Age = output
Range of outputs [min, max]	[1.764, 7.15]	[0.394, 6.74]	[0.18, 7.15]	[−69.70, 1.69]
Distribution of risk scores				
Boundaries	• No risk: less than 3 • Low risk: 3 to 4 • Moderate risk: 4.1 to 4.9 • Likes risk: 5 to 5.8 • High risk: more than 5.8	• No risk: less than 1.5 • Low risk: 1.6 to 2.3 • Moderate risk: 2.4 to 3.3 • Likes risk: 3.4 to 4.3 • High risk: more than 4.3	• No risk: less than 1.5 • Low risk: 1.6 to 2.3 • Moderate risk: 2.4 to 3.3 • Likes risk: 3.4 to 4.3 • High risk: more than 4.3	• No risk: less than −50 • Low risk: −49 to −40 • Moderate risk: −39 to −30 • Likes risk: −30 to −17 • High risk: more than – 17

After boundary class category distribution				
Total number of observations in each category	• No risk : 1 : 60,923 • Low risk : 2: 8,17,511 • Moderate risk : 3 : 15,15,986 • Likes risk: 4 : 6,90,604 • High risk: 5 : 60,701	• No risk : 1 : 9,96,032 • Low risk : 2: 11,76,069 • Moderate risk : 3 : 8,08,223 • Likes risk: 4 : 5,36,121 • High risk: 5 : 2,33,555	• No risk : 1 : 6,53,408 • Low risk : 2: 10,55,754 • Moderate risk : 3 : 11,18,259 • Likes risk: 4 : 6,37,694 • High risk: 5 : 2,84,885	• No risk : 1 : 4,22,859 • Low risk : 2: 7,17,505 • Moderate risk : 3 : 4,22,859 • Likes risk: 4 : 11,56,591 • High risk: 5 : 8,04,616
Final data sample chosen for models (stratified/ solving the imbalanced class problem)	**9,43,718 rows of data.** • No risk : 1 : 18,277 • Low risk : 2 : 2,45,254 • Moderate risk : 3 : 4,54,796 • Likes risk : 4 : 2,07,181 • High risk : 5 : 18,210	**11,25,000 rows of data.** • No risk : 1 : 2,98,810 • Low risk : 2 : 3,52,821 • Moderate risk : 2,42,467 • Likes risk : 4 : 1,60,836 • High risk : 5 : 70,066	**11,25,000 rows of data.** • No risk : 1 : 196022 • Low risk : 2 : 316726 • Moderate risk : 335478 • Likes risk : 4 : 191308 • High risk : 5 : 85466	**11,25,000 rows of data.** • No risk : 1 : 1,26,858 • Low risk : 2 : 2,15,251 • Moderate risk : 3 : 1,94,529 • Likes risk : 4 : 3,46,977 • High risk : 5 : 2,41,385
Correlations of variables with final category column	• Age 0.453045 • Dependents 0.374792 • Annual_income 0.404084 • Monthly_loans 0.232607 • Income_stability 0.302768 • Portfolio_status 0.103018 • Investment_obj 0.222040 • Duration_inv 0.109305 • Comfort 0.345110 • Behaviour 0.283973 • output 0.931334 • categories 1.000000	• Age 0.742770 • Dependents 0.516709 • Annual_income 0.109562 • Monthly_loans 0.053320 • Income_stability 0.214954 • Portfolio_status 0.014864 • Investment_obj 0.063204 • Duration_inv 0.036067 • Comfort 0.074225 • Behaviour 0.018715 • output 0.966746 • categories 1.000000	• Age 0.640698 • Dependents 0.485437 • Annual_income 0.109455 • Monthly_loans 0.371431 • Income_stability 0.213423 • Portfolio_status 0.040315 • Investment_obj 0.118964 • Duration_inv 0.043781 • Comfort 0.131617 • Behaviour 0.114555 • output 0.963646 • categories 1.000000	• Age 0.849599 • Dependents 0.119673 • Annual_income 0.110896 • Monthly_loans 0.088254 • Income_stability 0.318025 • Portfolio_status 0.003605 • Investment_obj 0.057762 • Duration_inv 0.004719 • Comfort 0.094796 • Behaviour 0.015175 • output 0.971413 • categories 1.000000

Appendix 3- Explaining the machine learning models

Logistic Regression

Logistic Regression is a commonly used statistical method for analysing and predicting data with one or more independent variables and one binary dependent variable; for example spam or not spam email classifiers, benign or malignant tumour detection. A logistic regression classifier tries to fit data according to a linear hypothesis function such as $Y = W(i)x(i) + B$ (Similar to a line equation) where Y is the dependent variable, X represents independent variables from 1 to n, B gives an error bias (negligible) and W is the weight assigned to each variable. W is an important value as it tells us the individual contributions of variables in determining Y, our target.

The independent variable is always binary, in our case there will be five logistic regression classifiers with their independent variables as 1 (Low Risk) or Not 1 (Not Low Risk), 2 or Not 2 and so forth till case 5 (High Risk). This format of multiclass classification is called 'one versus rest', the input sample is passed through all the classifiers and probability of the sample belonging to classes 1 to 5 is calculated and the highest probability class wins.

The interpretation of weights in logistic regression is dependent on the probability of class classification, the weighted sum is transformed by the logistic function to a probability. Therefore the interpretation equation is:

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right)=\log\left(\frac{P(y=1)}{P(y=0)}\right)=\beta_0+\beta_1 x_1+\cdots+\beta_p x_p$$

The log function calculates the odds of an event occurring.

$$\frac{P(y=1)}{1-P(y=1)}=odds=\exp(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)$$

Logistic regression is used over linear regression as completely linear model do not give output probabilities because it treats the classes as numbers (0 and 1) and fits the best hyperplane (for a single feature, it is a line) that minimizes the distances between the points and the hyperplane. In other words, it simply interpolates between the points, and we cannot interpret it as probabilities. A linear model also extrapolates and gives us values below zero and above one. Logistic regression is also widely used, interpretable and fits our use case relatively well.

Support Vector Machine (SVM) Classifier

A support vector machine finds an equation of a hyper-plane that separates two or more classes in a multidimensional space; for example if we consider a two dimensional space, this “hyperplane” will become a line dividing the plane on which the data lies into two separate classes. If the data is not linearly separable i.e. there is no clear line separating the classes (This happens in many cases; imagine two classes in the data forming concentric circles) then data can be transformed onto a different plane (say we view the concentric circles from z axis) it becomes a linearly separable problem again (imagine the points in the circle having different depth). After separating it we can transform it back to the original plane : this is done using a kernel function in SVM.

Support vector machines have become wildly popular due to their robust efficiency and high accuracy despite requiring very few samples to train. They have disadvantages especially when it comes to time and space complexity but the SVM algorithm along with its variations are being used commercially in face detection and protein fold predictions.

SVM for multiclass classification trains $n*(n-1)/2$ classifiers, where n is the number of classes in the problem. Therefore for our problem there will be 10 different classifiers each will choose permutations of classes as the binary dependent variable (Y) i.e. 1 or 2, 2 or 3, 1 or 4 and all others. During this, each classifier predicts one class instead of probabilities for each.

Interpreting the above is quite difficult, the benefit of a linear model was that the weights / parameters of the model could be interpreted as the importance of the features. But if the model is non-linear it would not work. Once we engineer a high or infinite dimensional feature set, the weights of the model implicitly correspond to the high dimensional space which isn't useful in aiding our understanding of SVM's. What we can do is fit a logistic regression model which estimates the **probability** of label Y being 1, given the original features. We use maximum likelihood estimation to fit the parameters of the logistic regression model, the technique is called Platt Scaling.

For our use case we use a kernel with interaction effects for learning hyperplane boundaries as our original equation used to generate data is correlated in a equation with interaction effects, but this adds some more complexity to the algorithm. The kernel with interaction effects can be written as $K(x, x_i) = 1 + ((xx_i))d$; where x is the input vector and x_i represents support vectors (hyperplane equations).

Decision Tree classifier

Decision trees belong to the family of tree based learning algorithms, they are widely used for supervised classification as they create precise, well defined and hierarchical decision boundaries for categorical and continuous data. This differs from classifiers that use a single separation boundary (or line) such as logistic regression by iteratively splitting the data into subparts by identifying multiple divisive boundaries.

The conditions that make these divisions try to ensure an absence of impurities in the populations contained by them; for example a condition that decision tree will make to describes a 'banana' could be in the sequence type = "fruit", colour = "yellow", shape = "crescent", spots = "true" this leaves no place for uncertainty or impurity. The algorithm stops when all classes are pure or there are no features left to divide upon.

Unfortunately such sharp dividing conditions are not always possible or may exceed certain time and space limitations in real life. Therefore when a clear separation of classes is not possible then we can have a stopping condition that tolerates some impurity (For example gini impurity measures quality of such splits by calculating the probability of an incorrect classification of a randomly picked datapoint).

The impurity itself can be calculated using a measure of randomness, **entropy:** $H = -p(x)\ log(p(x))$ or $-p\ log(p) - q\ log(q)$ where p = probability of success and q = prob of failure. Ideally H should be as small as possible.

For a dataset like ours with multiple features, deciding the splitting feature i.e. most important dividing condition at each step is a complex task, this feature should reduce the impurity through the split or one with gives the most information gain. **Information gain** at each node is calculated by the lowest entropy generated nodes by the split. Starting from the root node, you go to the next nodes and the edges tell you which subsets you are looking at. Once you reach a leaf node, the node tells you the predicted outcome. All the edges are connected by 'AND'. For example: If feature x is [smaller/bigger] than threshold c AND etc… then the predicted outcome is the mean value of y of the instances in that node.

Individual decisions made by the tree can also be explained by going down a particular path based on the input given. Decision trees can be used to explain the dataset by themselves.

Naïve Bayes

Naive Bayes classifiers are a family of classifiers that work on predicting future outcomes using conditional probability, given a history of behaviour. For example, given a year long history of weather forecasts with features such as humidity, rainfall, and temperature, a classifier from the naive Bayes family can be trained and used to predict future weather conditions. Due to its simplicity it has found a place in many real world systems such as credit scoring systems, weather prediction and many others. Given its popularity, we have used it to model our dataset.

The Bayes algorithm works under a "naive" assumption that all the features are independent in nature, in our case that means the naive Bayes classifier is going to assume that our variables such as age, income are uncorrelated so finding probabilities can be thought of as a simple counting calculation. This implies that the classifier won't be a right fit for our case as we know that the data was generated using many correlations (such as age will affect an individual's income, behaviour etc..).

If the naive Bayes classifier wants to calculate the probability of observing features f_1 to f_n, given a class *c* (In our case *c* here, represents the risk class and *f* values represent all our question-answer scores), then

$$p(f_1, f_2, \ldots, f_n \mid c) = \prod_{i=1}^{n} p(f_i \mid c)$$

This means that when Naive Bayes is used to classify a new example, the posterior probability is much simpler to work with:

$$p(c \mid f_1, f_2, \ldots, f_n) \propto p(c) p(f_1 \mid c) \ldots p(f_n \mid c)$$

But we have left $p(f_n \mid c)$ undefined i.e. the occurrence of a certain feature given a class which means we haven't taken the distribution of the features into account yet. Therefore for our case we have used a **gaussian naive Bayes** classifier that simply assumes $p(f_n \mid c)$ is a gaussian normal distribution, this works well for our data which is a normal distribution.

Then the formula for our low risk class used by the classifier will be something like:

*P(low-risk/Age, Income, Dependents ..) = P(low-risk/Age-category) * P(low-risk/Income-category) etc/P(Age) * P(income)* etc. This will be calculated for all risk categories and the class with the highest probability is given as the final prediction.

Naive Bayes is an interpretable model because of the independence assumption. It can be interpreted on the modular level. The contribution made by each feature towards a specific class prediction is clear, since we can interpret the conditional probability.

K-Nearest Neighbours (KNN)

Neighbours-based classification is a type of *instance-based learning* or *non-generalizing learning*: it does not try to construct a general internal model, but simply stores instances of the training data. In KNN, a data point is classified by a majority vote of its neighbours. The input is assigned the class most common among its 'k' nearest neighbours, where 'k' is a small positive integer, the value of 'k' is chosen depending on the data. KNN is very useful in applications that require searching for similar items;

such as recommender systems, bio-surveillance software, document retrieval systems such as concept search which is used in many e-Discovery software packages.

These neighbours are decided using brute force techniques that calculate distance from the data point of interest to all the other data points in the dataset, by using formulae like Euclidean distance. This means that the time and space complexity of this operation is very high; for *n* samples in *d* dimensions the time complexity will be $O(d*n*n)$ which makes this algorithm relatively slow to run on large datasets.

Since KNN is an instance based algorithm there is no learned model, there are no parameters to learn, so there is no interpretability on a modular level. There is a lack of global model interpretability because the model is inherently local and there are no global weights or structures explicitly learned. To explain a prediction at a local level, we can always retrieve the k neighbours that were used for the prediction. This is useful for our dataset as there will be thousands of neighbouring data points but presenting those 'k' nearest points could be a very useful explanation for each category.

Bibliography

Abraham, F., Schmukler, S. L., & Tessada, J. (2019, Febuary). *Robo-Advisors: Investing through Machines*. Retrieved October 2019, from http://documents.worldbank.org/curated/en/275041551196836758/text/Robo-Advisors-Investing-through-Machines.txt

Andrews, L. (2017). Algorithms, governance and regulation: beyond 'the necessary hashtags'. In LSE, *Algorithmic Regulation* (pp. 7-12). London.

Arner, D., Barberis, J., & Buckley, R. (2016). *FinTech, RegTech and the Reconceptualization of Financial Regulation.*

Baer, D. (2019, November). *The 'Filter Bubble' Explains Why Trump Won and You Didn't See It Coming.* Retrieved October 2019, from The Cut: https://www.thecut.com/2016/11/how-facebook-and-the-filter-bubble-pushed-trump-to-victory.html

BBVA. (2017, November 18). *What is a regulatory sandbox?* Retrieved December 2019, from https://www.bbva.com/en/what-is-regulatory-sandbox/

Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). *Machine learning explainability in finance: an application to default risk analysis.* https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2019/machine-learning-explainability-in-finance-an-application-to-default-risk-analysis.pdf, Bank of England.

Carey, T. (2019, September 24). *Investopedia*. Retrieved October 2019, from https://www.investopedia.com/robo-advisors-2019-where-have-all-the-assets-gone-4767826

Castelluccia, C., & Le Métayer, D. (March 2019). *Understanding algorithmic decision-making: Opportunities and challenges.* Study, European Parliamentary Research Services, Panel for the Future of Science and Technology.

Chazot, C. (2015, October). (R. E. INSTITUTE OF INTERNATIONAL FINANCE, Interviewer)

Ciocca, P., & Biancotti, C. (2018, October 23). Data superpowers in the age of AI: A research agenda. *VOX CEPR Portal*.

Citron, D. K., & Pasquale, F. A. (2014). The Scored Society: Due Process for Automated Predictions. *Washington Law Review, 89*, 1–34.

Costello, M., Hawdon, J., Ratliff, T., & Grantham, T. (2016, May). Who views online extremism? individual attributes leading to exposure. *Computers in Human Behavior*.

Cowls, J., King, T., Taddeo, M., & Floridi, L. (2019, May 15). Designing AI for Social Good: Seven Essential Factors. *SSRN https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3388669*.

Dastin, J. (2018, October 10). *Amazon scraps secret AI recruiting tool that showed bias against women*. (Reuters) Retrieved September 2019, from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Datta, A., Tschantz, M. C., & Datta, A. (2015, February 18). Automated Experiments on Ad Privacy Settings. *Proceedings on Privacy Enhancing Technologies*, 92–112.

Diakopoulos, N., Friedler, S., Arenas, M., Barocos, S., Hale, M., Howe, B., et al. (n.d.). *Principles for Accountable Algorithms*. Retrieved December 2019, from FAT ML: https://www.fatml.org/resources/principles-for-accountable-algorithms

EU GDPR. (2016). *EU GDPR Chapter 3.* Retrieved October 2019, from https://gdpr.eu/article-22-automated-individual-decision-making/

FINRA. (2016, March). *Report on Digital Investment Advice.* Retrieved September 2019, from FINANCIAL INDUSTRY REGULATORY AUTHORITY: https://www.finra.org/sites/default/files/digital-investment-advice-report.pdf

Goodman, B., & Flaxman, S. (2017). *European Union regulations on algorithmic decision-making and a "right to explanation"*. Retrieved October 2019, from https://ora.ox.ac.uk/catalog/uuid:593169ee-0457-4051-9337-e007064cf67c/download_file?file_format=pdf&safe_filename=euregs.pdf&type_of_work=Journal+article

Hall, P., & Gill, N. (2018). *An Introduction to Machine Learning Interpretability.* (N. Tache, Ed.) O'Reilly.

IOSCO. (2014, July). *Report on the IOSCO Social Media and Automation of Advice Tools Surveys.* Retrieved September 2019, from https://www.iosco.org/library/pubdocs/pdf/IOSCOPD445.pdf

Jenik, I., & Lauer, K. (2017). *Regulatory Sandboxes and Financial Inclusion.* CAGP. https://www.cgap.org/sites/default/files/Working-Paper-Regulatory-Sandboxes-Oct-2017.pdf.

Kapur, D., & Khosla, M. (2019). *Regulation in India: Design, Capacity, Performance.* Hart Studies in Comparitive Public Law.

Kari, P. (2019, October 25). *Healthcare algorithm used across America has dramatic racial biases*. Retrieved October 2019, from Guardian: https://www.theguardian.com/society/2019/oct/25/healthcare-algorithm-racial-biases-optum

Kaya, O. (2017, August 10). *Robo-advice – a true innovation in asset management.* Retrieved September 2019, from https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000449125/Robo-advice_%E2%80%93_a_true_innovation_in_asset_managemen.pdf

Laboure, M., & Braunstein, J. (2017, November 11). *Democratising finance: The digital wealth management revolution*. Retrieved October 2019, from VOX CEPR Policy Portal: https://voxeu.org/article/digital-wealth-management-revolution

Maurell, v. d. (2019). *Embracing Robo Advisory looks promising or the longitivity of Financial Advisors.* Global Financial Markets Institute, New York.

Mulgan, G. (2016, February). A machine intelligence commission for the UK: how to grow informed public trust and maximise the positive impact of smart machines. *Nesta*.

Narayanan, A. (2016, June 27). *Investor Business Daily.* Retrieved October 2019, from https://www.investors.com/etfs-and-funds/etfs/fund-industry-wakens-from-slumber-to-take-on-digital-advice-upstarts/

New, J., & Castro, D. (2018). *How Policymakers Can Foster Algorithmic Accountability.* Center for Data Innovation.

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019, October 25). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*.

Padmanabhan, A., & Rastogi, A. (2019). Big Data. In D. Kapur, & M. Khosla, *Regulation in India: Design, Capacity, Performance* (pp. 251-278). Hart Studies in Comparitive Public Law.

ProPublica. (2016, May 23). *Machine Bias There's software used across the country to predict future criminals. And it's biased against blacks.* Retrieved September 2019, from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August 9). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *arxiv*.

Rudin, C. (2019, May 13). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 206–215.

Sample, I. (2017, January 27). *This article is more than 3 years old AI watchdog needed to regulate automated decision-making, say experts*. Retrieved January 2020, from The Guardian: https://www.theguardian.com/technology/2017/jan/27/ai-artificial-intelligence-watchdog-needed-to-prevent-discriminatory-automated-decisions

SEBI. (2016, October 26). *Consultation Paper on Amendments/Clarifications to the SEBI (Investment Advisers) Regulations, 2013.* Retrieved August 2019, from sebi.gov.in: https://www.sebi.gov.in/sebi_data/attachdocs/1475839876350.pdf

SEBI. (2016, December 8). *SEBI (Investment Advisers) Regulations 2013 [Last amended on December 08, 2016]*. Retrieved 2019 August, from sebi.gov.in: https://www.sebi.gov.in/legal/regulations/jan-2013/sebi-investment-advisers-regulations-2013-last-amended-on-december-08-2016-_34619.html

Shneiderman, B. (2017, May 30). *Algorithmic Accountability*. The Alan Turing Institute.

Thelisson, E., Padh, K., & Celis, E. L. (2017, July 15). Regulatory Mechanisms and Algorithms towards Trust in AI/ML.

Tutt, A. (2016, March 15). An FDA for Algorithms. *Administrative Law Review*.

Wachter, S., Mittelstadt, B., & Floridi, L. (2016, December 28). Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation. *International Data Privacy Law 2017*.

Wired.com. (2019, November 19). *The apple card didn't see gender and that's the problem*. Retrieved December 2019, from https://www.wired.com/story/the-apple-card-didnt-see-genderand-thats-the-problem/

Acknowledgements

We would like to thank Tarunima Prabhakar, Prof. Mark Hallerberg, reviewers from the National Institute of Securities Market, the blind reviewers, and Viraj Kulkarni for their feedback and suggestions on the paper.

About the Authors

Sanjana Krishnan

Sanjana is a partner at CPC Analytics. As part of CPC, she has consulted for public sector and nongovernmental organizations such as GIZ, Swiss Agency for Development and Cooperation and Save the Children, among others. She volunteers with DataMeet, an open data community working towards open, accessible and usable government data. Sanjana has a Masters in Urban Policy and Governance from Tata Institute for Social Sciences, Mumbai, an undergraduate degree in physics from St. Xaviers College, Mumbai, and a diploma in Cyber Law from Government Law College, Mumbai.

Sahil Deo

Sahil is a co-founder of CPC Analytics. As a part of CPC, he has consulted leading private and public sector organizations such as Siemens, Mitsubishi Materials, the World Health Organization and GIZ (German International Cooperation). He has also worked on a European Union project in Morocco & Egypt, which aimed to create financially viable models for investment in the renewable energy sector. As a part of Civic Consulting, he was a part of the team studying household indebtedness in member countries for the European Commission. Sahil has a Master of Public Policy from the Hertie School of Governance, Berlin and an undergraduate in computer engineering which he pursued at the University of Pune, India and Ecoles des Mines, France. He is currently pursuing his PhD from Hertie School of Governance, Berlin.

Neha Sontakke

Neha works as a research analyst and data scientist with CPC Analytics. Her past experience includes building artificially intelligent systems in the domains of text analytics, sentiment analysis, entity recognition as well as recommendation systems. She has a Bachelors of engineering in computer science from Pune Institute of Computer Technology.

🌐 datagovernance.org ✉ dgn@idfcinstitute.org

@datagovnetwork /datagovnetwork /datagovnetwork