protiviti®
Global Business Consulting

ML AND VARIANCE BIAS TRADEOFF

Face the Future with Confidence®

TABLE OF CONTENTS

SECTION 1

EMERGENCE OF MACHINE LEARNING IN THE LANDSCAPE OF RISK MANAGEMENT

Machine Learning (ML) models are enabling financial institutions to make more accurate and efficient risk management decisions compared to the traditional modelling algorithms, since they are designed to learn directly from data, identify patterns without constant manual intervention. It has been observed that in recent years many global banks have implemented ML-based models in different areas of risk management like credit scoring, collections strategies, fraud detection, developing trading strategies by detecting anomalies in the behavioural patterns of the customers from huge volumes of structured and unstructured data.

The training of the model to learn the patterns or objects from the data distributions and is termed as "fitting process". It involves teaching a computer algorithm to make predictions on new unseen data, by recognizing patterns in historical datasets. The training of a model requires an objective function (which specifies the purpose for which the model is trained), which if not met, the model repeatedly trains itself. The retraining of the model allows the ML system to improve over time resulting in higher predictive accuracy and efficiency.

The Machine Learning development pipeline has been summarized in the diagram below:

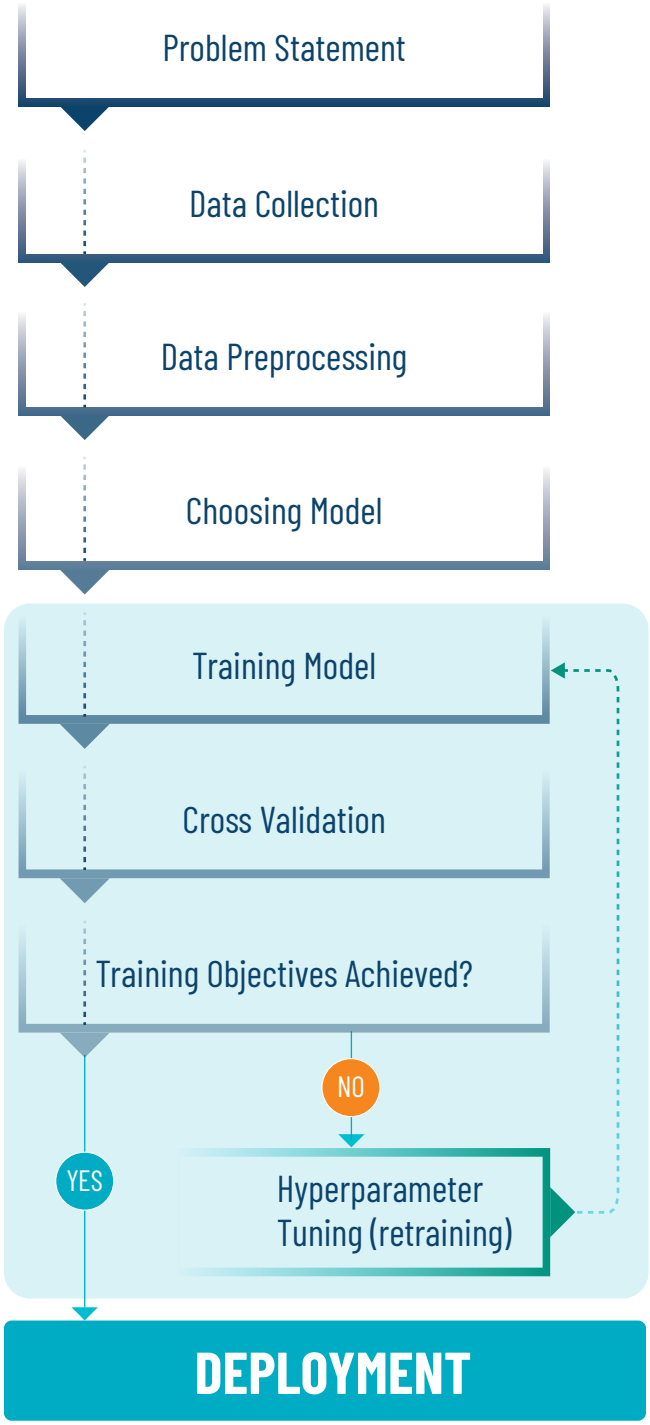


Figure 1: *ML Process*

Risk Management practices have been leveraging ML models to capture complex patterns and non-linear relationships between parameters to accurately determine the key sources of risks to which the Banks and Financial Institutions (FI) are exposed. Despite the advantages and opportunities that ML models unleash, they come with their own shortcomings. One such shortcoming is the problem of Variance Bias trade-off, which adversely impacts the model's generalization performance (the ability of a machine learning model to perform with precision on an unseen dataset). Bias refers to the problem of model underfitting (where the model misses most important patterns in the data) that occurs due to over-simplistic assumptions used in the learning algorithm. For example: A Linear model is used to fit a non-linear data. On the other hand, Variance refers to the error that arises due to sensitivity to small fluctuations in the training data set which results in over-fitting the data and thereby, capturing the noise along with the information. The core challenge of Variance Bias trade-off is that reducing bias (by introducing complexity in the model) increases the variance, and on the other hand, reducing the variance (by simplifying the model) increases the bias. The concepts of Bias and Variance in a Machine Learning model is described in Figure 2 and Figure 3 below:

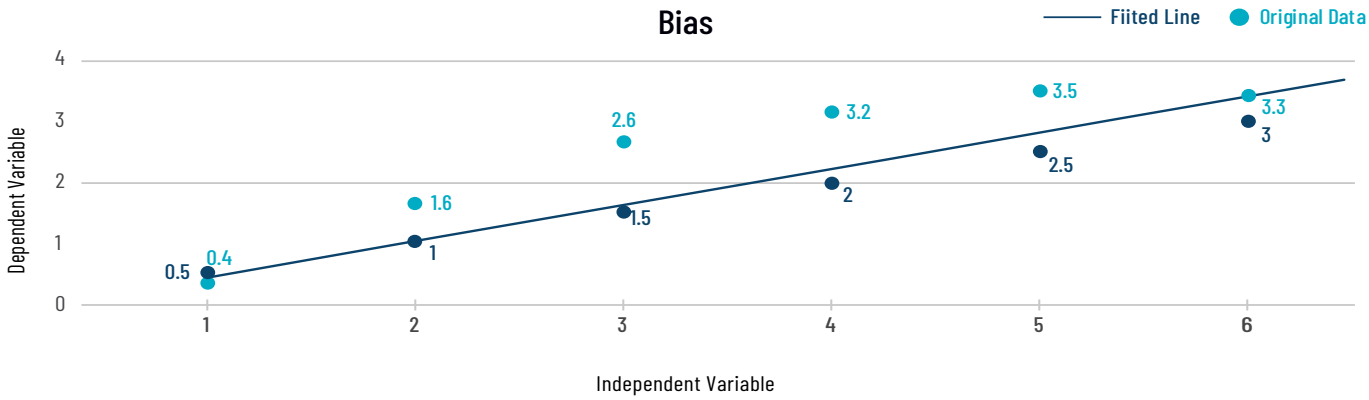


Figure 2: *Bias*

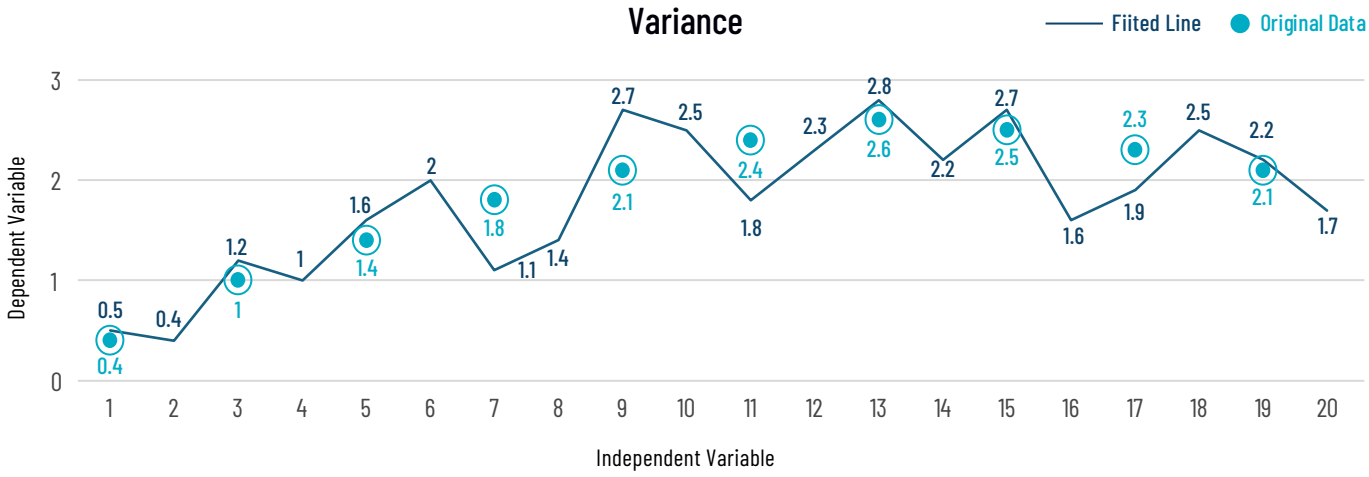


Figure 3: *Variance*

The paper is divided into the following sections: Section 2 and Section 3 discusses the kinds of bias that is observed in a Machine Learning model and how they should be interpreted and the concept of Variance and its minimization techniques respectively, Section 4 explains the variance bias trade off in detail, Section 5 focusses on the methods of addressing the Variance Bias Tradeoff and Section 6 concludes.

SECTION 2

INTRODUCTION TO BIAS

Bias refers to the error that is introduced by approximating a complex real-world problem with a simplified model, leading to underfitting. It implies that the model makes strong assumptions about the data and does not capture the underlying relationship between input and output accurately. Bias also refers to the unfair or discriminatory outcomes produced by the model, resulting from biased training data or model design. Therefore, to ensure that the outcomes of a Machine Learning model are accurate, generalized, and fair, it is important to identify the various sources from where bias can arise and minimize the impact of biases on the final model outcome. Bias, in a Machine Learning model, can arise from any of the three key stages of a model development life cycle mentioned below:

1. **The input stage from training data.**
2. **The processing stage from algorithms.**
3. **The output stage from predictions by the model.**

SECTION 2.1

BIAS AT THE INPUT STAGE OF TRAINING DATA (INPUT BIAS)

At the Input Stage where the data is being constructed to train the Machine Learning model, bias can arise due to societal inequalities and prejudices that are embedded into the data distributions (Historical Bias), heterogeneity in the nature of representing certain groups in the training dataset (Representation Bias), the skewness in data collection or interpretation due to the influence of prior beliefs (Confirmation Bias), inconsistent and misleading relationships in the data distribution as a result of the omission of an important predictor (Omitted Variable Bias), bias in the target variable due to historical factors or subjective human labelling (Label Bias) etc.

Historically, it has been observed that Bias in Machine Learning models has negatively impacted fair-lending norms in credit lending, resulting in discriminatory practices and regulatory violations. The cornerstone of Fair Lending is that the obligors should be risk ordered in terms of their risk profiles and should not be discriminated based on factors like: Race, Gender, Age, Religion, National Origin, Marital Status etc. If the historical loan data distribution reflects past discrimination, the model may learn to reproduce similar patterns, even if the explicit variables are dropped from the data resulting in an unintended bias. There are many instances where Financial Institutions have faced lawsuits, fines, and reputational damages because of the presence of unintended biases in lending practices.

For example: In 2022, Wells Fargo encountered allegations of discriminatory lending practices that arose due to the design of an algorithm to assess the creditworthiness of borrowers. The model had assigned higher risk scores to Black and Latino applicants with respect to White applicants; hence the Black and Latino population experienced a high rate of loan rejection[1].

Therefore, it is important to identify the presence of historical bias in the data distributions before building the model. A key measure to quantify the bias (or fairness) is a Disparate Impact (DI) which is the ratio of favourable outcomes (e.g. Loan approvals) between protected (groups that are legally or ethically protected from discrimination) and Reference groups (Majority or historically favoured groups). As DI → 1.0, there are no disparate impacts since the Selection Rate for the Protected Group is equal to the Selection Rate for

1. Case Study: Algorithmic Bias in Loan Denials | by Bashir Iliyasu Bashir | Medium

the Reference Group. Similarly, if DI → 0 the potential impact against the Protected Group increases, since the Selection Rate for the Protected Group becomes close to zero. There are regulatory standards (like U.S. Equal Employment Opportunity Commission) which specify D.I < 0.8 as evidence of adverse impact of historical bias.

Bias in Machine Learning models can also arise from non-random selection of the data samples and non-representative target population. This bias, known as the **Sampling Bias,** refers to a type of systematic error that arises when the training datasets are not representative of the actual population distribution and there is an underrepresentation of some groups, while for other groups there is an overrepresentation. Under such a situation, the model learns patterns that do not align accurately to the actual target population. Sampling Bias can lead to inaccurate predictions, unfair outcomes and reduced generalisation.

As a result, the models trained with an inherent sampling bias can perform well on the training data but would have high chances of failing when it is implemented on real-time data. Further, the model would have fairness issues where it would be overfit on dominant groups, and the minority or protected groups would receive unfair predictions.

For example: a credit scoring model trained on the affluent segments may under predict the creditworthiness for applicants with lower income. Models trained with Sampling Bias can lead to skewed predictions and overfitting to dominant patterns, leading to feedback loops (since ML use a trained model as an input to develop a better model) and worsening long-term bias. Therefore, it is important to identify the existence of Sampling Bias in a Machine Learning model. A very popular metric to quantify sampling bias is the Kullback-Liebler (KL) Divergence, which is a statistical measure of how one probability distribution (distribution of the training data) diverges from a second, reference distribution (population distribution, test or production data distribution, distribution across sub-groups). Mathematically, the KL Divergence is represented as follows:

$D_{KL}(P/Q) = \sum_x P(x)\log\left(\frac{P(x)}{Q(x)}\right)$; where $P(x)$ = The reference population distribution,
$Q(x)$ = The observed distribution from the sample.

The severity of the sampling bias is directly proportional to the value of the KL Divergence measure. A large value of KL implies that the sampling bias is large since the divergence of the observed distribution is large from the reference population. Similarly, a small KL value implies that the training sample is close to the true distribution.

Closely related to Sampling Bias, is the Representation Bias which arises when the data used to train the model does not adequately represent the real-world diversity of the population it is intended to model. It is type of a sampling bias but is more focussed on the skewness of the feature distributions in the data. This bias arises from factors such as historical exclusions (where the data is based on records where certain populations are excluded or ignored), skewed data sources (where the data is collected from certain regions, platforms or demographics) and under-representation of edge cases (where low-frequency classes, minority groups are not well covered).

Representation Bias can lead to reduced accuracy for under-represented groups since the model may perform poorly on those classes that are not well represented. This can further lead to fairness violations in the decisioning process of the model use case, where a disparate impact on the protected or the minority group can be identified. In the long run, the model built with representation bias can reflect or amplify societal biases as well. Therefore, it is important to identify the existence of Representation Bias in a Machine Learning model. For identification of Representative Bias, a multitude of non-parametric statistical measures which detect imbalances, under-representation or distributional differences between different sub-groups of the data are used.

For example: Kolmogorov-Smirnov (KS) test, Anderson-Darling (AD) test, Jenson-Shannon (JS) Divergence test. The KS test compares the Cumulative Distribution function of a one-dimensional continuous variables between two samples by calculating the maximum absolute difference between the two Cumulative Distribution functions. A large value of the KS statistic suggests that there is a Representative bias since the two distributions are not the same. A better statistical test to assess the differences in the distributions due to under-representation is the Anderson-Darling (AD) test, since it is more sensitive than the KS test especially in the tails of the distribution. A large value of the AD test statistic suggests that there is an underrepresentation for the continuous variables under consideration. These statistical tests help in detecting representation

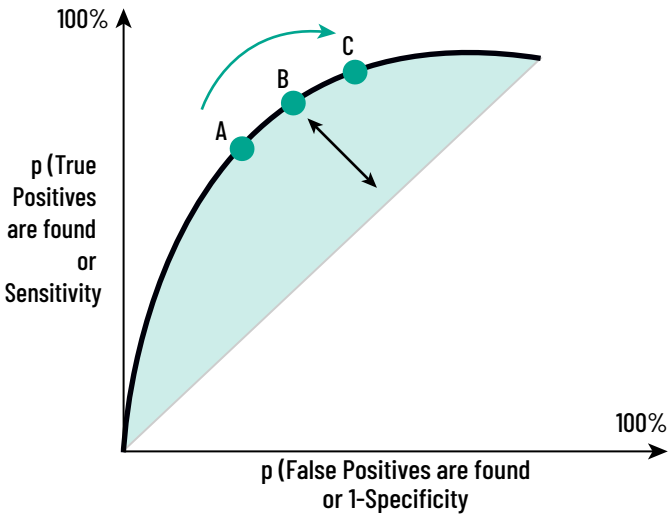
bias by revealing whether the dataset accurately and appropriately reflects the diversity and structure of the population under consideration.

The other important types of input bias include **Omitted Variable Bias, Label Bias and Conformity Bias.** Omitted Variable Bias arises when a relevant feature (variable) is left out of the model or dataset, and that missing variable is correlated with both the input and output which results in spurious correlations that arises when the model compensates by incorrectly attributing effects to included variables. A typical case of Omitted Variable Bias is observed in underwriting scorecards, if demographic characteristics such as Sanction Flags for nations are excluded, which can lead to potential cases of terrorist funding, as the lending of the credit can get biased towards the affluent but an extremist customer. Label bias, on the other hand, arises when the distribution of the training labels (dependent variable) is biased because of historical prejudices, which can reinforce estimation bias in the model. Finally, Conformity Bias occurs when model developers or annotators unconsciously align their labelling or decisions with prevailing group opinions or prior models, thereby perpetuating existing trends or biases without critical evaluation. Together, these biases can compound, leading to unfair or inaccurate model predictions. This emphasizes the important of rigorous data audits, transparent modelling practices, and ongoing fairness assessments throughout the ML lifecycle.

SECTION 2.2

BIAS ARISING FROM THE ALGORITHMS USED DURING THE TRAINING STAGE (ALGORITHM BIAS)

Algorithm or Model Bias, in a ML model refers to the systematic errors or unfair outcomes that arise due to the way the algorithm itself is structured, trained or optimized. It is introduced through the design and functioning of the model or the model's learning process. A very important reason for the Algorithm Bias to occur is the tendency of the model's learning process to inadvertently favour certain outcomes or patterns. Other reasons for which Algorithm Bias can arise include **Model architecture Bias** (where the algorithms have inherent limitations in capturing specific relationships in the data resulting in skewed predictions), Optimization Bias (where the bias results from an incorrect design of the loss function or the optimisation procedure, such that they are unable to account for fairness or other ethical considerations), **Fairness and Equity Bias** (where the algorithm unintentionally favours certain groups over others, leading to unfair or discriminatory outcomes). To identify the extent of algorithm bias, the tendency of the algorithm to create performance disparities across sensitive sub-groups should be analysed. The test data is divided into sub-groups using sensitive variables like age, gender, nationality, race etc. For each group, the model performance measures like: Area under the Curve (AUC), F1-Score and False Positive Rate (FPR) are computed for each group. The measure of the algorithm bias is quantified through the Sub-group Performance gap (Δ). A summary of the key model performance measures used for algorithm bias identification is described below:



AUC Curve

- **Area under the Curve (AUC)**: The Area under the Curve (AUC) is used widely to analyze the discriminatory capacity of binary classification models. It is the area under the Receiver Operator Characteristics (ROC) curve, which represents the trade-off between the True Positive Rate (Proportion of actual positive cases correctly identified as positive) and False Positive Rate (Proportion of actual negative cases incorrectly identified as a positives). The AUC summarizes the overall performance of the classifier across all thresholds. As AUC → 1 it indicates

better separability between classes, meaning the model can better distinguish between positive and negative cases. The Sub-group performance gap (ΔAUC) can be defined as the Relative difference in the AUC of the Sub-group. Higher the Relative Difference in the AUC of the sub-group, greater is the extent of the algorithm bias. Unequal True Positive Rates and False Positive Rate across the sub-groups suggests bias. Similarly, discrepancies in True Positive Rates indicate unequal opportunity between the sub-groups. Empirically, it has been observed that if the shift of the AUC is more than 10%, the extent of algorithm bias is very significant.

- **F1-Score**: A statistical metric used to evaluate the performance of a classification model with imbalanced classes. The score is calculated by considering Precision and Recall which are functions of True Positive Rates (TPR) and False Positive Rates (FPR). Unequal TPR or FPR across groups suggests bias and Discrepancies in TPR indicate unequal opportunity. High Variation in False Positive Rates (FPR) and False Negative Rates (FNR) across groups implies that the model is making more mistakes for some groups than others. Calculation of the F1-score is explained below:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives + False Positive (TP+FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives + False Negatives (TP+FN)}}$$

$$\rightarrow \text{F1 - Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision + Recall}}$$

(Harmonic Mean of Precision and Recall.

In the absence of algorithm bias, the F1-score should be similar across the sub-groups.). F1-score can be used to create a bias estimate:

$$\text{Bias Gap} = |\text{F1 - Score}_{\text{subgroupA}} - \text{F1 - Score}_{\text{subgroupB}}|$$

If the Bias Gap measure exceeds 10%, the extent of algorithm bias is very significant.

Algorithm bias poses multiple challenges to machine learning models like generating unfair outcomes across groups, skewed model performance and feedback loops where the model may perform well at the overall level but can perform extremely poorly for specific sub-groups. Such biased outcomes can feed into future data, thereby reinforcing the bias over time.

SECTION 2.3

BIAS ARISING FROM THE MODEL OUTPUTS (OUTPUT BIAS)

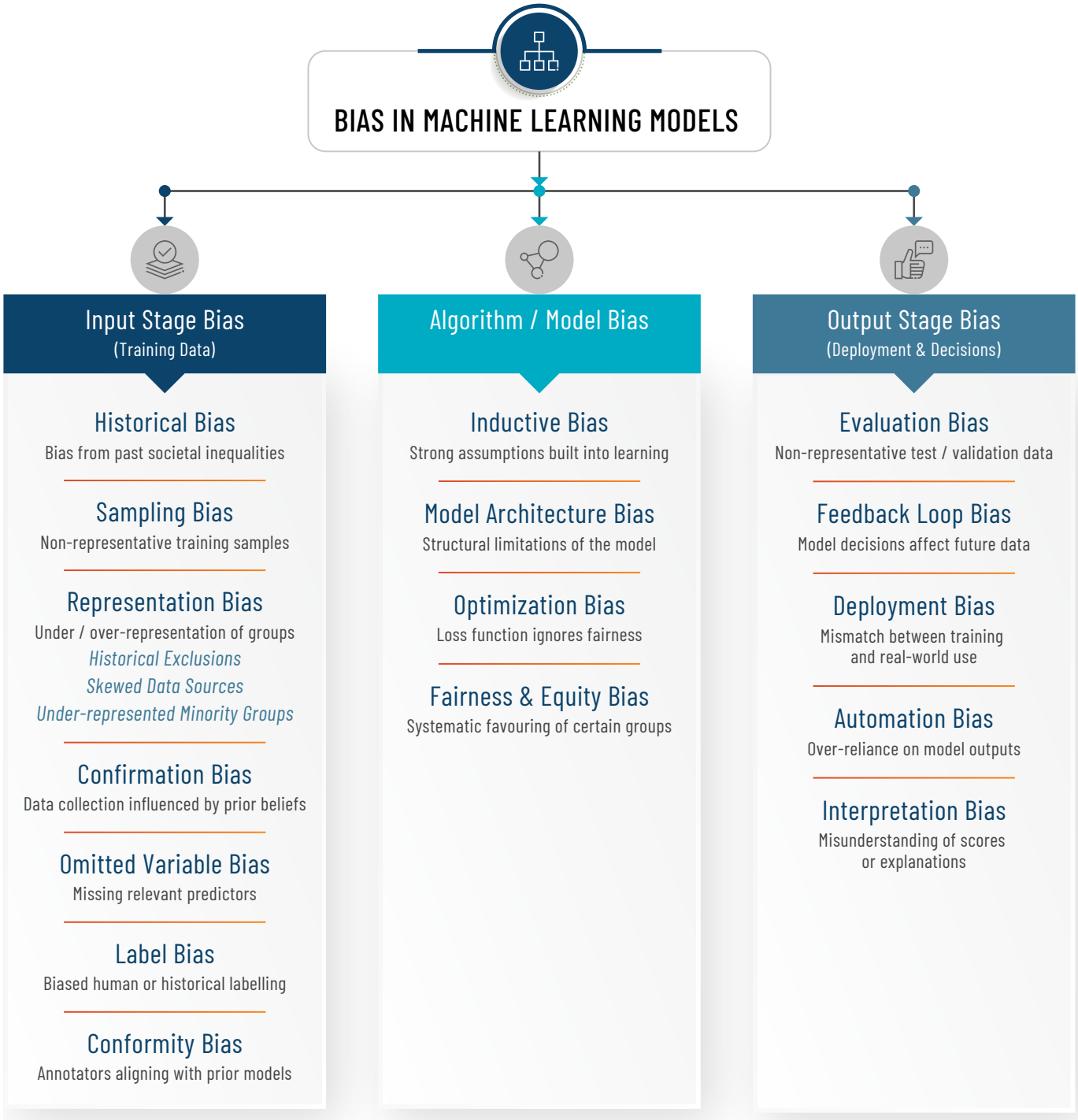
Output Bias, in an ML model refers to the distortions or unfairness in the predictions or decisions made by the model. This bias occurs often arises due to the model's inability to generalize well across different groups, the presence of bias in the earlier stages of the pipeline, or imbalances in how the model treats certain outcomes. This bias arises from factors like: Incorrect assignment of Decision Threshold (where the model can overpredict one class or underpredict another class), Imbalanced output distribution (where the model predicts one class disproportionately compared to others, especially in the case of imbalanced classes), Model Overfitting and Underfitting (where the model can learn from patterns that do not generalize well or the model cannot learn enough from the data which can also result in poor performance), post-processing factors (where some adjustments or calibrations are performed which can have disproportionate effects on certain groups). Output Bias has been historically observed in different areas of risk management. When Machine Learning models, trained on historical data (price, volume, news, indicators) are used to design trading strategies, the outcomes can be biased in favour of certain specific sectors or geographies and the model can consistently recommend trades in highly liquid assets, missing out profitable opportunities in the less liquid assets.

Similarly, in the space of Credit risk management, scorecards have biased lending decisions based on race, gender and nationality. One such case was when it was found that the parameter determining the location of the applicants was causing the model to price people in areas with Black and South Asian ethnicity higher in majority of the Credit Decisioning models in the US[2].

2. The FCA Consumer Duty: algorithmic bias and discrimination | Insights - BDO

Another critical form is **Feedback Loop Bias,** where model decisions influence the future availability of data, particularly labels. In credit lending, repeatedly rejecting applicants from specific demographics (e.g., low-income or new-to-credit individuals) can result in a lack of repayment data for those groups. Consequently, the model remains uninformed about how such applicants would have behaved, reinforcing a cycle of biased predictions and limiting its learning capacity. Identification methods include tracking how model decisions affect label availability, monitoring trends in reapplication rates and missing labels over time and using delayed outcome analysis to detect long-term feedback distortions.

TYPES OF BIASES IN THE MACHINE LEARNING LIFECYCLE



Bias can originate at any stage of the ML lifecycle and may compound if not mitigated.

Deployment Bias occurs when the model is applied in real-world settings that differ significantly from the training context. For example, deploying a model trained on salaried employees to assess gig or informal workers without recalibration can result in poor risk estimations. Techniques to detect deployment bias include measuring distribution shifts using metrics like the Population Stability Index (PSI), conducting Out-of-Distribution (OOD) tests, and using shadow modelling to simulate the model's behaviour in the deployment environment before going live.

Automation Bias refers to the over-reliance of human decision-makers on model outputs without critical judgment or contextual awareness. In practice, this manifests when loan officers accept model predictions without question, even when there are clear red flags in documentation or borrower behaviour. Automation bias can be identified by analysing override patterns, surveying users for model understanding and dependence, and auditing justification logs for manual interventions.

Finally, **Interpretation Bias** arises when model outputs are misunderstood or miscommunicated during decision-making. This includes situations where decision-makers misinterpret risk scores, fail to grasp the model's limitations, or misapply feature attributions. Interpretation bias can be mitigated by assessing the clarity of training materials, interviewing users to evaluate their understanding of the model, and implementing explainability tools like SHAP or LIME to enhance transparency.

Output Stage Biases present significant risks to the fairness, accuracy, and trustworthiness of machine learning systems, especially in regulated domains such as credit risk. These biases can erode model performance and equity even after rigorous training and validation procedures. Proactively identifying and mitigating these biases is crucial for ensuring that models behave reliably and ethically in real-world scenarios.

These output biases can lead to unjust credit decisions, disproportionately affecting marginalized or unconventional borrower groups. Therefore, to manage the extent of output bias, the underlying data and modelling practices should be scrutinized.

SECTION 3

INTRODUCTION TO VARIANCE

Variance, in a Machine Learning model refers to how much the model's predictions would change if the model were trained on different samples from the same distribution as the training data. A Machine Learning model is considered to have High Variance, when minor changes in the training data lead to large changes in the model outcomes. Similarly, when small changes in the training data leads to insignificant changes in the outcomes in different samples, the machine learning model is considered to have Low Variance. Models with low variance generalizes well to unseen data.

Variance measures how much, on average, predictions vary for a given data point:

$$\boldsymbol{Variance(x)} = \boldsymbol{E}[(\boldsymbol{f(x)} - \boldsymbol{E}[\boldsymbol{f(x)}])2]$$

SECTION 4

VARIANCE BIAS TRADE-OFF

The Variance and Bias errors cause the learning model to produce inaccurate results.

Mean-squared error can be calculated as:

$$Error = Bias\hat{}2 + Variance + Noise$$

Whose mathematical equation is:

$$Error(x) = (E[f(x)] - (f(x))2 + E[(f(x) - E[f(x)])2] + Noise$$

In case the model is underfitted, i.e., the bias is high, it will be observed that the training and the test data will generate low accuracy scores.

For models that are overfitted, i.e., the variance is high, the training set's accuracy would be high, but these models will perform significantly worse in the test data set, causing incorrect prediction.

Model developers need to find the optimal level of complexity where both the errors of bias and variance can be tackled. From the following validation curve let us see how once can achieve this optimal complexity which is termed as the variance-bias tradeoff.

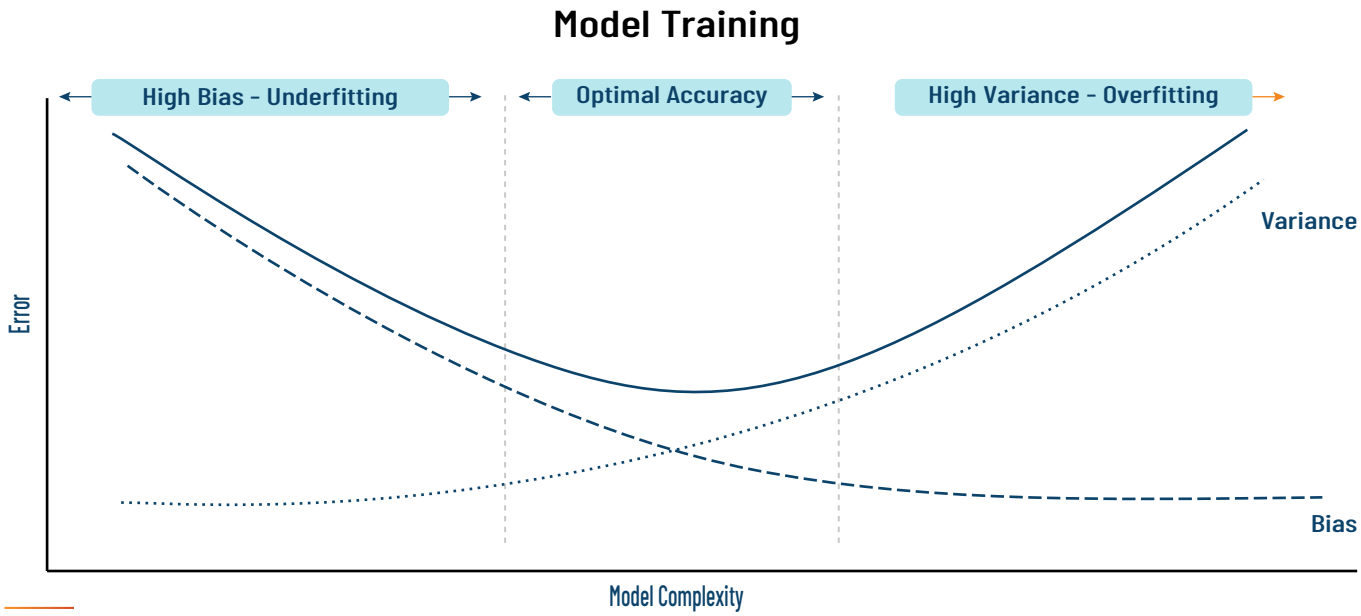


Figure 4: *Variance Curve*

From the above curve the three distinguished regions are:

High Bias

Underfitting where low model complexity is present, and both training and testing dataset show high errors.

High Variance

Overfitting where the model complexity is very high, and this starts fitting the noise along with the underlying patterns. Hence the model gives high accuracy on the training set, but on the validation set the accuracy score significantly drops.

Optimal Accuracy

where model generalizes well to training and validation datasets.

SECTION 5

MITIGATING THE VARIANCE AND BIASES IN THE MODEL – FINDING THE RIGHT BALANCE

Accurate predictions from a machine learning (ML) model rely on effectively managing the **bias-variance trade-off**. High **bias** results in underfitting, where the model fails to capture underlying patterns. While high **variance** results in overfitting, where the model learns noise rather than signal. Several techniques are used to strike the right balance between these two, improving generalizability and reliability of model predictions:

Regularization techniques like **L1 (Lasso), L2 (Ridge)** and Elastic Net can help to reduce overfitting and check variance by adding penalty terms to the model's cost function, preventing it from becoming overly complex. It is calculated as

$L_{reg} = L_{orig} + \lambda.\Omega(\beta)$ where L_{orig} is the original loss (e.g., MSE or Log Loss), β represents model coefficients, λ is the regularization parameter and Ω(.) is the penalty function.

Regularization improves linear regression by adding a penalty term to the standard regression equation. Lasso adds the **absolute value of magnitude** of the coefficient as a penalty term, which penalizes the absolute size of regression coefficients. It reduces some regression coefficients to exactly zero, which implies that it performs a feature selection and effectively simplifies the model by removing unimportant variables. Ridge Regression works by minimizing **the sum of squared differences** between the observed and predicted values by fitting a line to the data. There are many instances where it is observed that features which show high multicollinearity might be present in the model building. The penalty function in Ridge Regression (L2-Penalty) discourages large weight. By penalizing large coefficients, Ridge regression makes the model less complex and more stable across the different datasets. Ridge Regression does not reduce any coefficient to zero and it retains all features, but it reduces the influence of the variables and hence controls variance. Elastic net adds both the **absolute norm of the weights** as well as the **squared measure of the weights** to control the noisy features. While it helps with controlling the variance, but in the downside, it can increase bias hence finding the right regularization parameter λ is crucial for a robust model. There are some additional methods that can be used for reducing both Bias and Variance:

Feature engineering and a thorough feature selection process can easily reduce both bias and variance. Features or parameters selected to train a model can introduce bias. If features are selected without considering their impact on fairness (like considering gender as in important feature having an impact on the target variable, i.e. default rates) the model may inadvertently favour certain groups and introduce discrimination. Hence, optimal feature selection can help eliminate such bias. On the other hand, including multiple features that are highly correlated with each other in model development can introduce redundancy and noise, leading to unstable model coefficients and increased variance. To address this, **Variance Inflation Factor (VIF)** analysis is employed, where a high VIF value (calculated as

$$VIF_j = \frac{1}{1-R_j^2}$$

indicates the pre sence of multicollinearity. Additionally, **SHAP (Shapley Additive Explanations)** analysis can be used alongside VIF to identify which of the correlated features contributes most to the model's predictions. By removing less informative or redundant variables and retaining only the feature with the highest explanatory power, the model becomes more robust, interpretable, and less prone to overfitting.

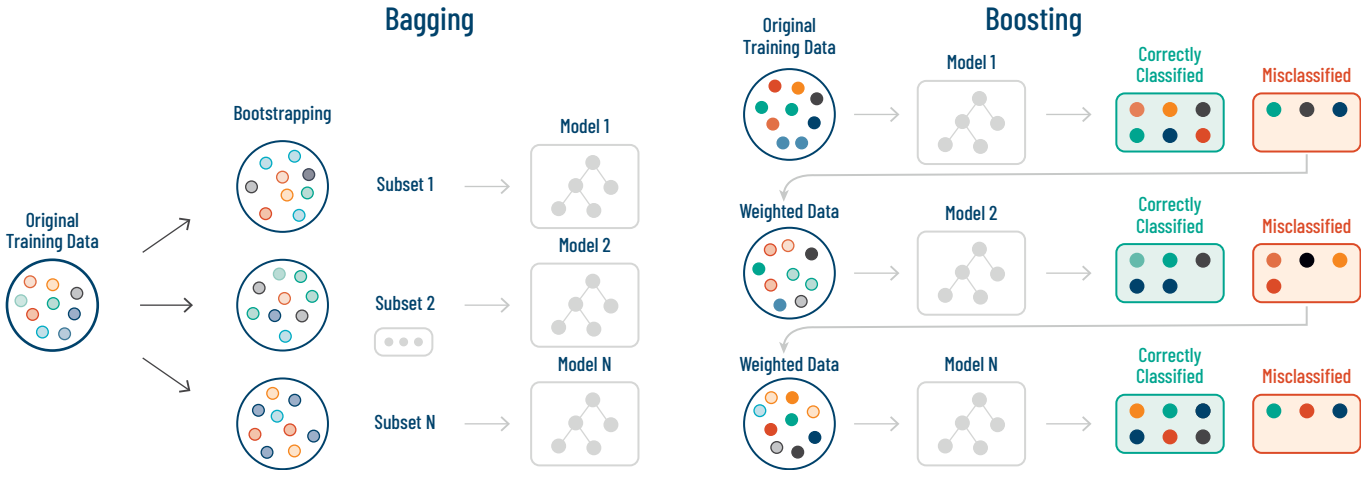
Another important method used for mitigating the Variance and Bias in a Machine Learning model is the **Feature transformation**. It involves applying various techniques to modify features in ways that make them more suitable for model learning such as: **Standardization and scaling** (e.g., min-max normalization or z-score standardization) to bring all attributes to a comparable range; **Log transformations** to reduce skewness and approximate a normal distribution for the variables; **Encoding categorical**

variables (e.g., one-hot or label encoding) for effective handling by algorithms and **Binning** to convert continuous variables into discrete intervals, simplifying patterns for the model. These transformations help mitigate issues like outliers, skewed distributions, and inconsistent scales, which can otherwise introduce **bias** and lead to poor generalization. Again, by reducing data noise feature transformation enhances both model fairness and predictive performance. Therefore, it helps in controlling both **bias** and **variance** in the model.

Cross-validation is another technique for evaluating a model's performance across different subsets of the data. The most common type is K-**Fold Cross-Validation** where the dataset is divided into K equally sized folds. The model is trained on K-1 folds and validated on the remaining fold. This process is repeated K times, with each fold serving as the validation set. By training and validating the model on multiple splits, cross-validation ensures that the model's performance is not dependent on a particular subset of the data. This helps in reducing high variance, where a model might perform well on training data but poorly on unseen data. However, it's important to note that when **K is small** (e.g., K=2 or 3), the size of each training fold is relatively small, which can lead to an **increase in bias**. This happens because the model is trained on fewer data points in each iteration, which may prevent it from learning the full complexity of the data. As a result, the model becomes too simplistic, leading to **underfitting**. Conversely, using a **very high K** (as in **Leave-One-Out Cross-Validation**, a type for K fold cross validation where each training set differs by just one data point) helps reduce bias, since the model is trained on nearly the entire dataset in each iteration. However, this can lead to **higher variance** in the evaluation, because the test error is based on only one data point at a time, making it very sensitive to outliers or noise. Therefore, choosing an appropriate *K* is crucial for achieving a balanced bias–variance trade-off that ensures both model stability and generalization.

Ensemble Methods are also very useful and are widely used for reducing Variance Bias Trade-offs in developing Machine Learning models. It combines the predictions of multiple models to produce a more robust and accurate overall model. Two of the most widely used ensemble methods are **Bagging and Boosting**. **Bagging** helps reduce variance by smoothing out the predictions across multiple models. It also prevents overfitting to the noise in any one sample, without oversimplifying the model, as each model still captures meaningful patterns from the data, thus maintaining a balance between both bias and variance reduction. **Boosting**, by focusing more on the hard-to-predict instances, reduces bias of the model. Over time, it builds a strong learner that can capture more complex patterns, reducing systematic errors (i.e., bias) that a single weak model would make. Bagging primarily helps reduce variance, while Boosting mainly helps reduce bias, and both contribute to building more accurate and stable machine learning models.

To summarize, achieving the right balance between bias and variance is essential for building an effective and reliable machine learning model. By accurately identifying whether a model is underfitting or overfitting and implementing appropriate corrective measures, its performance can be significantly optimized. Throughout the model development process, it is important to recognize various sources of bias and mitigate the risks they pose at every stage. Ultimately, selecting the right algorithm and using appropriate and correct data tailored to the specific problem is critical for delivering an optimal and high-quality ML system.



Bagging and Boosting

MANAGING THE BIAS-VARIANCE TRADEOFF IN MACHINE LEARNING

Managing bias-variance trade-off affdes the accurate and reliable models to achieve accurate and reliable Ensemble Methods.

Regularization

- **Lasso, L1, Ridge:** Elastic Net
- **Ridge (L2):** Reduces some coefiicients to zero
- **Elastic Net:** Combines L1, L2 regularization

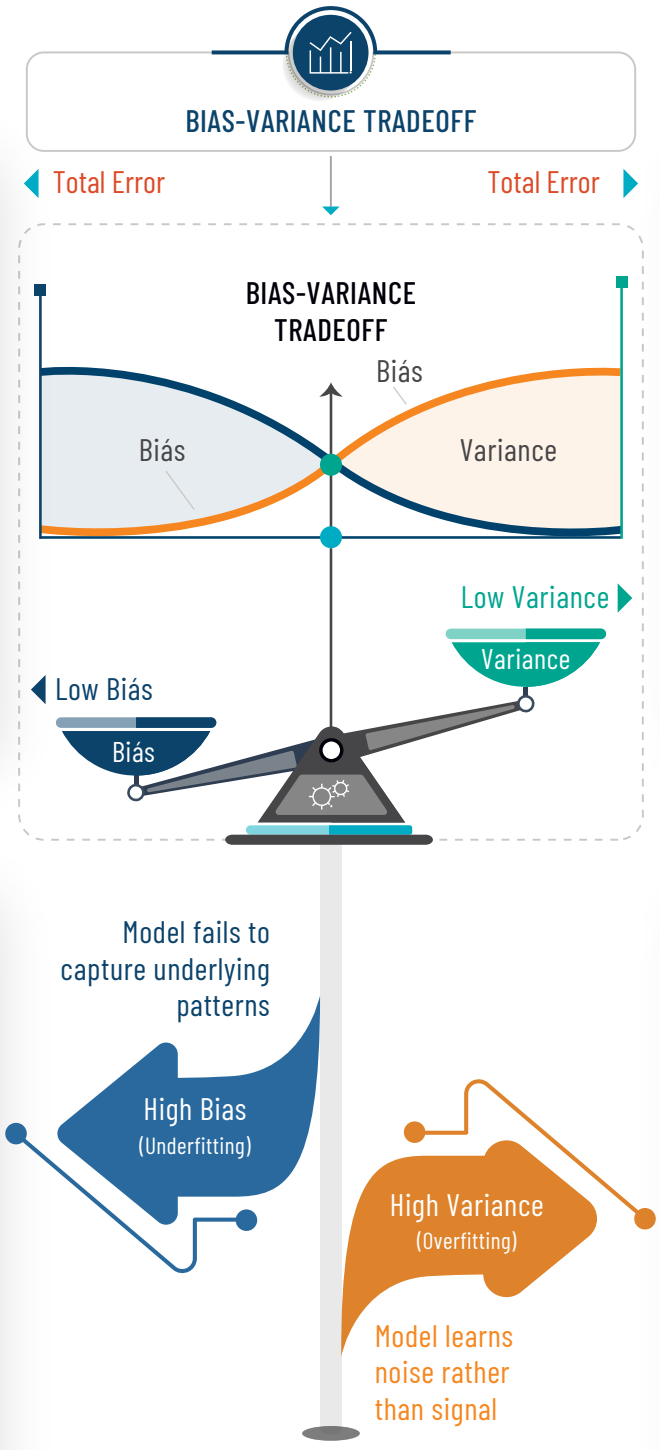
$$L_{reg} = L_{orig} + A.O.(B)$$

Reduces overfitting by preventing overly complex models

Feature Engineering

- Optimal feature selection & transformation
- Variance Inflation Factor (VIF)
  - Identifies multicollinearity
- **SHAP Analysis:** Determine key predictors

Reduces both bias and variance by selecting and transforming critical features



Ensemble Methods

- **Bagging** (*Bootstrap Aggregating*)
  - Reduces variance
- **Boosting** (Adaptive Boosting)
  - Reduces bias

Combines multiple learners to improve predictive performance

Cross-Validation

- **K-Fold** Cross Validation: Splits data into K subsets
- **Leave-One-Out:** Trains on all data but one point, tests on the left-out point
- **Stratified CV:** Preserves class distribution across folds

Evaluates model performance across different data folds

Balancing these techniques helps achieve accurate and reliable model predictions.

SECTION 6

CURRENT MARKET CHALLENGES

ML-based credit scorecards offer opportunity, but unmanaged bias and fairness risks can outweigh the benefits unless tightly governed and assessed by appropriate risk management practices. In the present day and age of financial inclusion, financial institutions are increasingly focussing on women-led households and enterprises, rural borrowers, and informal-sector businesses which were historically underrepresented in formal credit systems. Therefore, traditional credit modelling frameworks which were built on decades of legacy retail and corporate banking and bureau data, are insufficient and non-representative of the distribution characteristics of the present obligor base. The historical datasets are inherently biased towards the underrepresented classes, and the behaviour and credit characteristics are very different from the current borrower base resulting in high model bias, poor generalisation and inaccurate predictions. Since Fair-lending expectations are varying across markets and geographies and is a still evolving space, therefore, it is imperative that model biases be addressed since, unmanaged bias and fairness risks can expose the institution to regulatory, reputational, and strategic harm.

Mitigating Bias in a ML model faces multiple challenges in the real world. A primary problem arises from the limited access to the protected attribute data since the regulators often prohibit collecting sensitive demographic data which limits the bias detection exercise reliant on only a few proxy variables. As a result, the bias cannot be fully measured or proven leading to increased uncertainty on the model output. Another problem arises from the heterogeneity in the Fair lending standards across different jurisdictions and regulators which increases the localization costs and slows cross-market deployment. Finally, the most important challenge with trying to mitigate bias is that removing or constraining biased features can reduce model accuracy, resulting in resistance from business teams and slow adoption of advanced ML models.

Mitigating bias in ML-based credit scorecards is as much a market and governance challenge as a technical one, constrained by regulation, cost, explainability, and trust considerations.

SECTION 7

HOW CAN PROTIVITI HELP?

Protiviti serve clients using its expertise in machine learning (ML) by helping them develop models that strike the right balance between bias and variance, which is critical for creating accurate, reliable, and generalizable ML solutions by applying advanced regularization techniques, cross validation, ensemble models and Bayesian techniques.

Protiviti assists clients in creating complete, accurate and reliable data for developing Machine Learning models, which reduces the noise, build meaningful features that reduce model complexity, use dimensionality reduction techniques to improve model interpretability and reduce overfitting.

Protiviti team comes with strong technical ML expertise, deep domain knowledge, a strong understanding of governance and risk management which enables the clients to get an accurate, compliant and robust ML solutions, tailored to their business goals.

ABOUT PROTIVITI

Protiviti (www.protiviti.com) is a global consulting firm that delivers deep expertise, objective insights, a tailored approach and unparalleled collaboration to help leaders confidently face the future. Protiviti and its independent and locally owned member firms provide clients with consulting and managed solutions in finance, technology, operations, data, digital, legal, HR, risk and internal audit through a network of more than 90 offices in over 25 countries.

Named to the Fortune 100 Best Companies to Work For® list for the 11th consecutive year, Protiviti Inc. has served more than 80 percent of Fortune 100 and nearly 80 percent of Fortune 500 companies. The firm also works with government agencies and smaller, growing companies, including those looking to go public. Protiviti Inc. is a wholly owned subsidiary of Robert Half (NYSE: RHI).

Contacts

Nancy Bhatt
Managing Director
+91 7045652906
Nancy.Bhatt@protivitiglobal.in

Amrita Das
Senior Consultant
+91 8017920185
Amrita.Das@protivitiglobal.in

Tanmoy Ganguli
Associate Director
+91 9836850743
Tanmoy.Ganguli@protivitiglobal.in

Acknowledgement

Tanmoy Ganguli, Associate Director & **Amrita Das**, Senior Consultant from Risk and Compliance, has contributed to this publication.

PROTIVITI INDIA OFFICES

Ahmedabad
6th Floor, West Gate, E-Block,
Near YMCA Club, SG Highway,
Gujarat, 380 015, India

Bengaluru
Umiya Business Bay - 1, 9th Floor
Cessna Business Park, Outer Ring
Road, Kadubeesanahalli, Varthur
Hobli Bengaluru – 560 049
Karnataka, India

Bhubaneswar
1st Floor, Unit No 104, 105, 106
Utkal Signature, Chennai Kolkata
Highway Pahala, Bhubaneswar
Khordha - 752 101
Odisha, India

Chennai
10th Floor, Module No. 1007
D Block, North Side, Tidel Park
No. 4, Rajiv Gandhi, Salai,
Taramani, Chennai - 600 113
Tamil Nadu, India

Coimbatore
TICEL Bio Park, (1101 - 1104)
11th Floor Somaiyapalyam
Village, Anna University Campus,
Maruthamalai Road, Coimbatore
North Taluk, Coimbatore – 641046
Tamil Nadu, India

Gurugram
15th & 16th Floor, Tower A,
DLF Building No. 5, DLF Phase III
DLF Cyber City,
Gurugram – 122 002
Haryana, India

Hyderabad
Q City, 4th Floor, Block B,
Survey No. 109, 110 & 111/2
Nanakramguda Village
Serilingampally Mandal,
R.R. District
Hyderabad – 500 032
Telangana, India

Kolkata
PS Srijan Corporate Park,
Unit No. 1001 10th & 16th Floor,
Tower - 1, Plot No. 2
Block - EP & GP Sector-V,
Bidhannagar Salt Lake
Electronics Complex
Kolkata –700 091,
West Bengal, India

Mumbai
1st Floor, Godrej Coliseum
A & B Wing Somaiya Hospital Road
Sion (East) Mumbai – 400 022
Maharashtra, India

Mumbai - Goregaon
The Westin Garden City,
13th Floor, Commerz
1- International Business Park,
Behind Oberoi mall, South Side,
Goregaon, Mumbai - 400063,
Maharashtra, India

Noida
Windsor Grand, 14th & 16th Floor
1C, Sector - 126 Noida
Gautam Buddha Nagar- 201313
Uttar Pradesh, India

Protiviti India Member Private Limited

152294_Oct25

Face the Future with Confidence®