# Submitted by – Mohammad Tayyab Alam 24157

# Kaggle Competition Learning Summary

The kaggle completion involved predicting customer overstress in a microfinance setting while experimenting with a pool of 11 machine learning models like Decision trees, logistic , xgboost, stacking , catboost, lgboost, adaboost, naïve bayes, knn, random forest and gradient boosting.

Throughout this competition I submitted around ~72 models and experimented with different versions of each of these machine learning model types. This competition allowed me to learn the following concepts and their importance

1. Hyper parameter tuning: I experimented with several versions of **XGBoost, CatBoost, LGBoost**, randomforest models by tuning their models via randomsearchcv of the python to find the best hyper parameter. For xgboost for instance which was my initial model was somewhere around 0.92 but when I tuned the parameters the accuracy went up to 0.94.
2. Feature engineering: I also learned about the importance of feature engineering. Throughout the competition I experimented with different ways to break the 0.94 barrier of accuracy. I learned that engineering meaningful variables is more important than complex models as when I used feature engineering of sum, median based features my accuracy jumped to 0.95!
3. Out of fold predictions (OOF): This was my first time using OOF predictions, where the model is trained on all folds except one and tested on the excluded fold. I learned that this approach gives a much more reliable estimate of real-world performance since the validation data remains unseen during training, helping me detect overfitting early and improve model generalization.
4. Regularization : During hyper parameter tuning I also realized the importance of this option. Without it all of my models were in the 80% range but using this increased to 90% mark as it removed noisy and irrelevant features that added no value.

I approached the problem by dividing my challenge into two phases:

* **Phase 1**: I tested 10+ models including Decision Tree, KNN, Logistic Regression, and Naive Bayes. These **basic models failed** due to high dimensionality (79 features) and severe class imbalance (only 0.26% positives).
* **Phase 2**: I focused on advanced models like **XGBoost, CatBoost, LGBoost, and Stacking**, which performed significantly better thanks to their robustness, ensemble nature, and ability to handle imbalance and nonlinear patterns. This experience taught me **why tree-based ensemble models outperform simpler ones** in real-world datasets.

Overall, this competition taught me the importance of feature engineering and hyper parameter tuning to better solve imbalanced problems of predicting fraud, classification or overstress. By using my learnings I was able to secure a 7th position on the leaderboard with an accuracy of 0.95338 using XGBOOST model with feature engineering , missing values imputed using median and class imbalance handled.