Credimi OpenDataPlayground - Funicello

The notebook submitted is not comprehensive of the full effort spent trying to climb the leader board scattred around 10 notebooks; I've explored the data multiple times and tried multiple approaches, it has been a great opportunity to get my hands dirty and learn.

The best performance was obtained through the use of the ***LGBMclassifier*** with Bayesian Hyper-parameter Optimization using ***BayesianOptimization*** and ***scikit-optimize*** packages. Source for the functions snippets is given in the notebook.

The Bayesian Optimization searches the best hyper-parameters in the given set by running for 50 steps starting from 50 random points, behaviour set through the `n_iter` and `init_points` parameters in the `maximize` function of the `BayesianOptimization` object. The optimization is run over a 3 fold `Stratified Cross Validation` scoring the runs through a custom evaluation metric `credimiMetric`.

The `credimiMetric` is composed by the `confusion_matrix` function that returns the instances of each classification group of the binary confusion matrix. The predictions probabilities in input are thresholded by a threshold of 0.2 decided through empirical experience on the problem. The metric score is then calculated by normalizing on the possible achievable range to obtain a 0-1 score.

The model is then trained with the best found hyper-parameters on a 10 `Stratified KFold` for 15000 rounds with the `early_stopping_rounds` set to 250 on the evaluation function `credimiMetric`. The final probability predictions for the submission test set are mean for each row of the predictions given by the best iteration models for each kfold run.

An additional possible development that has not been explored is finding the optimal metric threshold, that is currently hardcoded, through cross validation.

Other notable approaches I tried that weren't performing as good or just didn't stick:

- Creating additional time period related features from the `dt_rif` feature like `month`, `quarter`, `year`
- Creating additional financial equilibrium KPI features from domain knowledge: `PFN`, `PFN/PN`, `EBITDA/PN`
- Creating yearly percentage growth features of `EBITDA` and `revenues` through their slope values
- Training on oversampled (SMOTE) and undersampled databases using the `imblearn` package
- Creating k-means clusters for each debt class on the 3 behaviour features to use as features instead of slopes
- Feature selection through Chi-Squared metric and MI for categorical variables
- Feature selection through Maximal Information-based Nonparametric Exploration using the `minepy` package to find non-linear relationships on numeric features