Bank Marketing Project

Introduction

This project is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. Thus given a set of client specific features, the aim of this ML project is to ascertain whether a given customer would subscribe to a bank sponsored financial product (term deposit in this case). If the trained ML model can have a high recall , the bank can better utilize its marketing budget and efforts in targeting only those clients , who have a high probability of subscribing or buying the financial product.

Data Wrangling & Cleaning

The original dataset has a total of 20 features (both numerical & categorical features) as well as a response variable and is highly imbalanced with the ratio of class ‘0’ or ‘not subscribed’ to class ‘1’ or ‘subscribed’ being 7.8760. The dataset is quite clean and didn’t require much cleaning except for standardizing the column names, removing duration feature as discussed below and converting categorical variables to dummies & removing one dummy per category to avoid linear dependence between the resulting dummy features, which automatically took care of missing values as well.

Exploratory Data Analysis

Missing Values Analysis

All the missing values are present in the categorical variables, with the missing values in housing (housing loan) and loan (personal loan) columns being perfectly correlated, as depicted in the plots 1 & 2 below below.

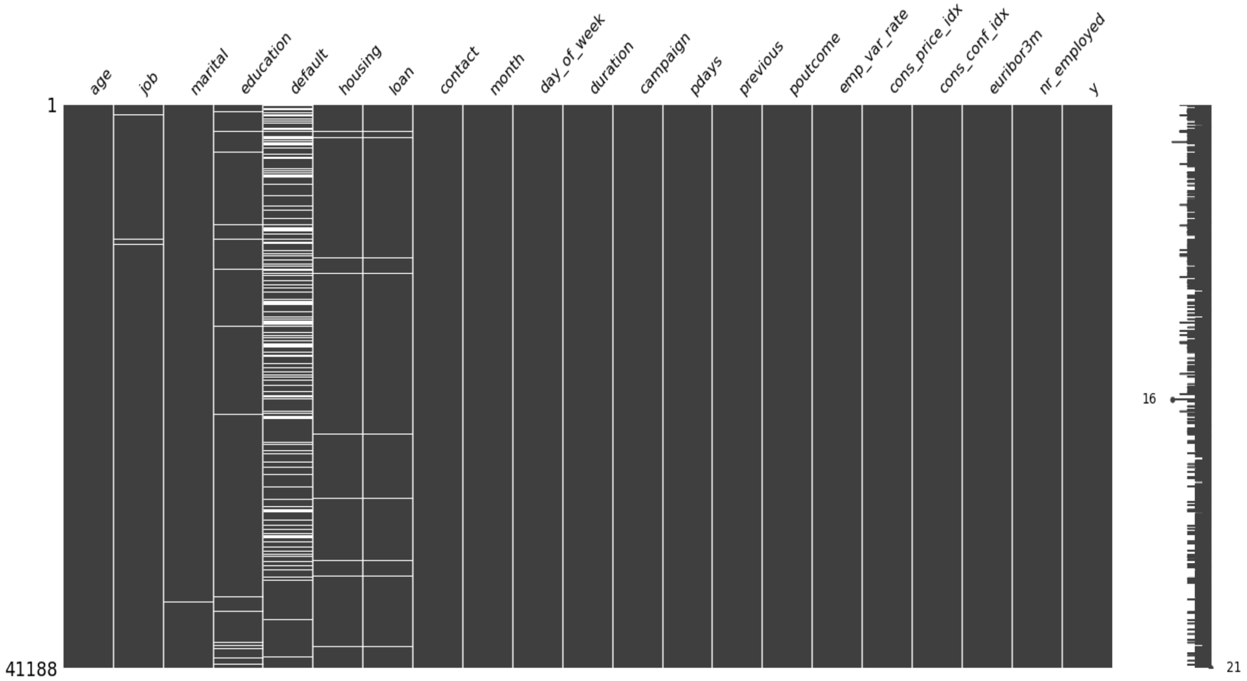


Fig_1: Missing Values Plot

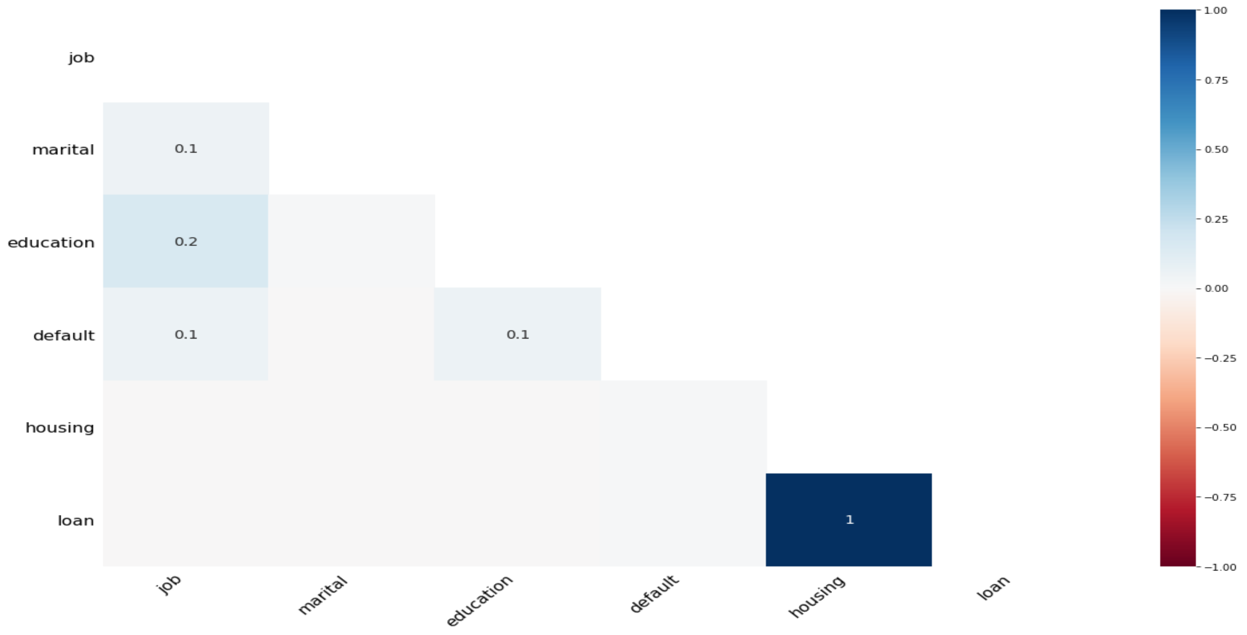


Fig_2: Missing Values Correlation

Also, there is clear evidence of multi-collinearity (fig_3) being present in the dataset, with various socio-economic variables being highly collinear with each other. Hence the linear models such as Logistics Regression may not give the best results on this dataset.

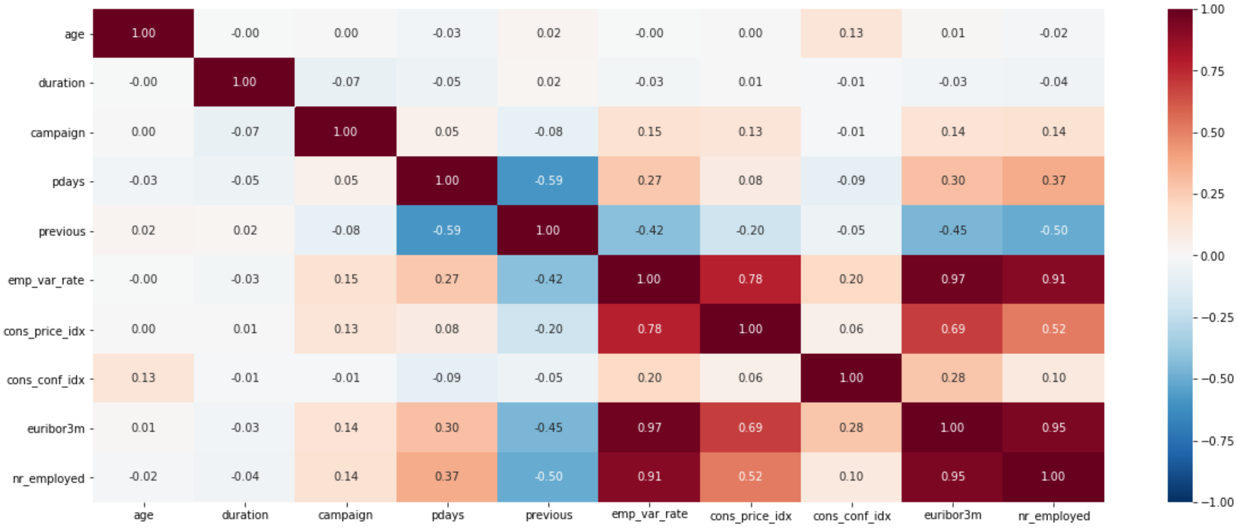


Fig_3: Multi-Collinearity between numerical columns.

Moreover a detailed examination of various features resulted in the following interesting observations.

a) The duration of the last call tended to be comparatively shorter for Non-subscribers than that for the subscribers. Also if (duration = 0) => y=0 almost surely. Also after a call, the outcome is more or less known. Thus this variable is certainly leaking information and should be removed from the feature list, in order to build a realistic predictive model.

b) Further comparatively more % of customers who were contacted again, subscribed to term deposit than the totally new ones. May be re-contacting more customers from the previous campaigns would have resulted in better success (fig_4).

c) Except for various Socio-Economic variables, all other numerical columns have outliers.

d) Among various job profiles, comparatively high % of retired & students who were contacted, subscribed to the term deposit (fig_5).

e) More than 50% of subscribers had a previous housing loan (fig_6).

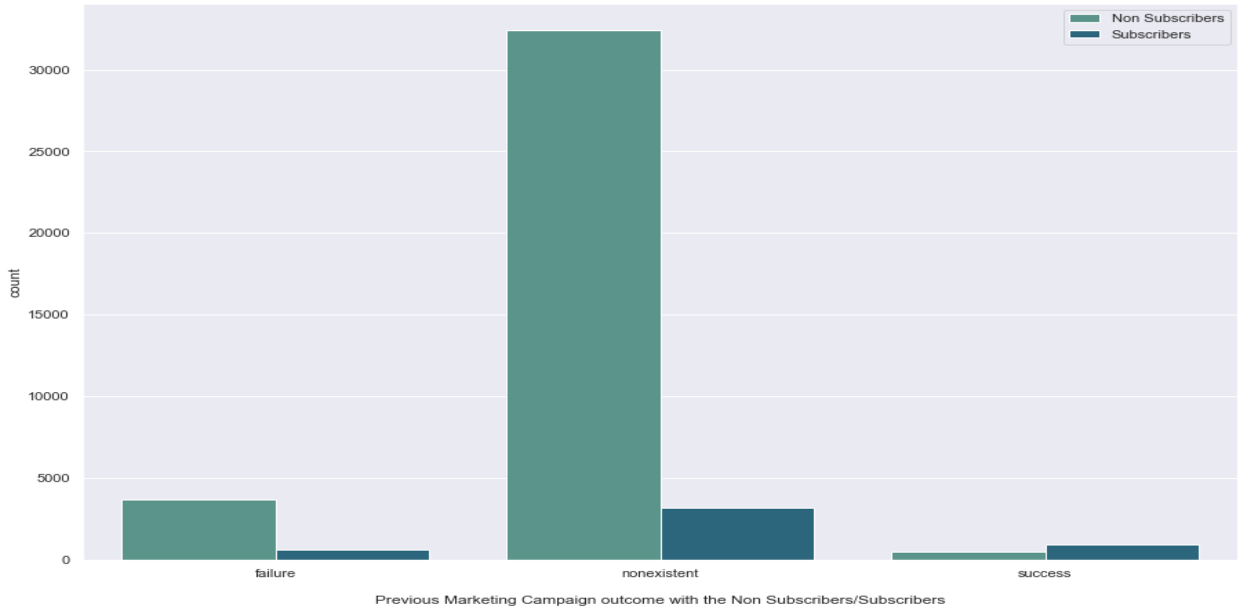


Fig_4.

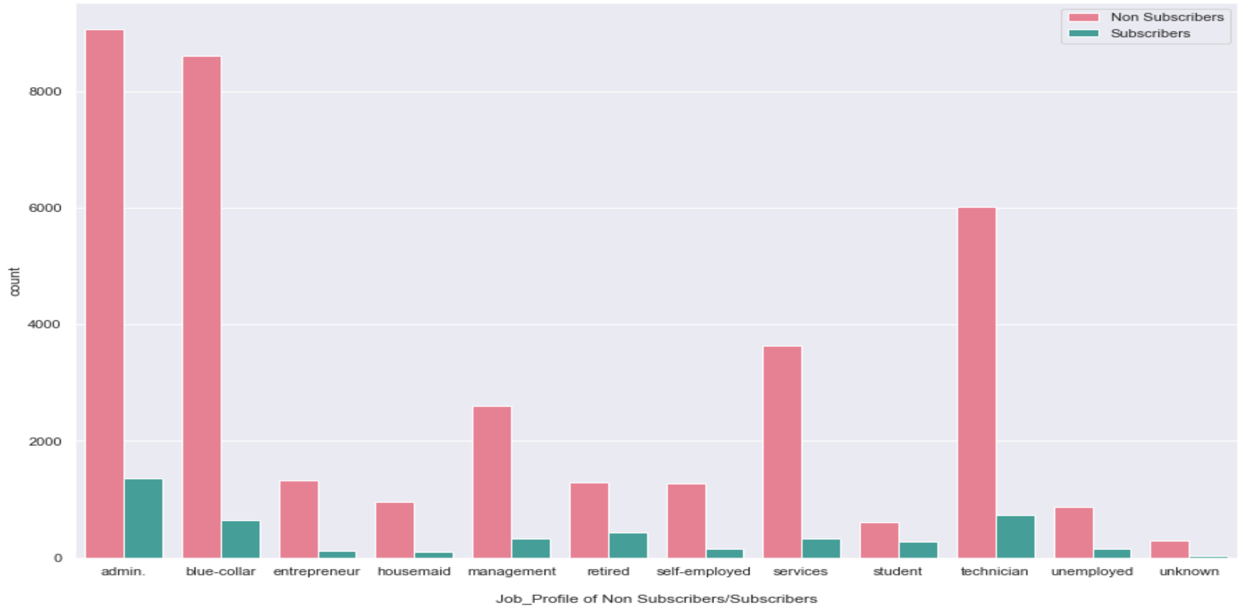


Fig_5.

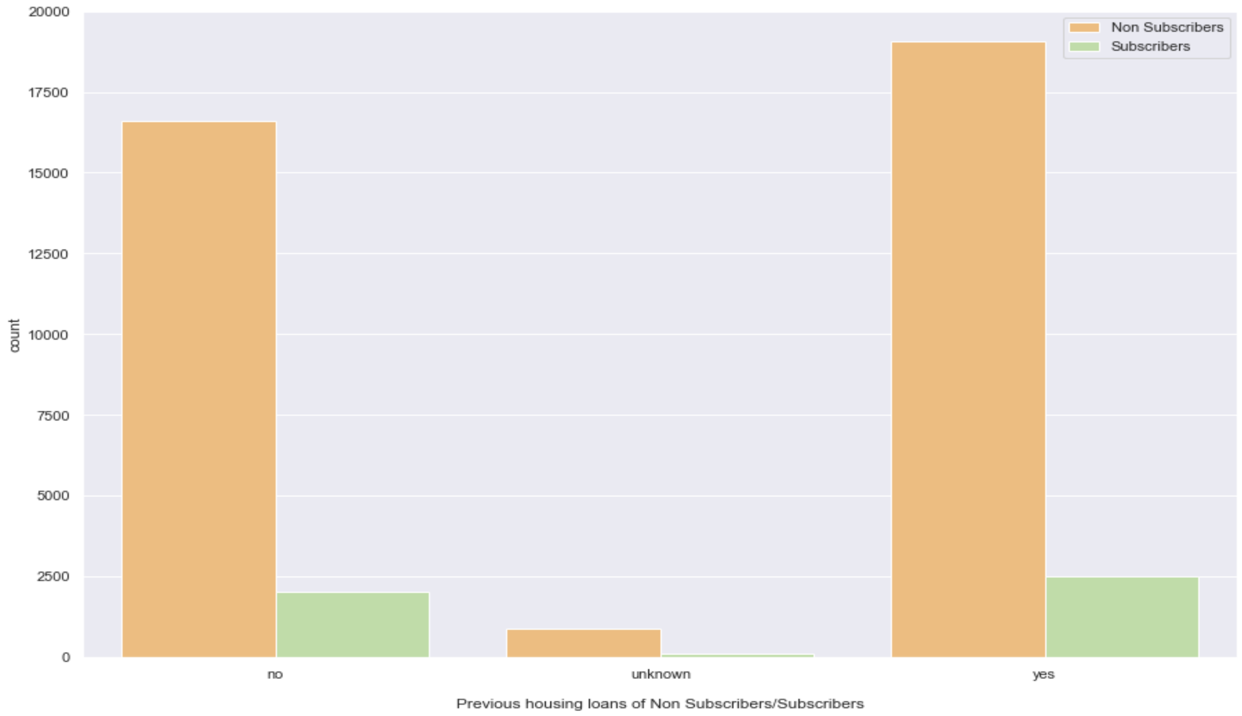


Fig_6.

Statistical Analysis

Since the dataset is highly imbalanced, I have chosen test set roc_auc score as the metric of choice. As mentioned previously, after cleaning the dataset, I was able to zero in on the “duration_ of_last_call” feature, which was found to be leaking information and hence was removed in order to build a realistic predictive model having some statistical forecasting power.

Further, there is clear evidence of multi-collinearity being present in the dataset, with various socio-economic variables being highly collinear with each other. This un-necessarily increases the dimensionality of the problem without providing any commensurate rewards. Hence, the linear models such as Logistics Regression would certainly not produce the best results on this dataset. But other models such as Random Forests, Light Gbm etc. would not be much affected.

In view of the above analysis, it was pertinent to reduce the dimensionality of the feature space. For this I chose the Mutual Information (**MI**) between the response variable y and each of the features X_i to facilitate the same. Mutual information is a measure of the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. Further it takes both linear as well as nonlinear dependencies between two RVs into account unlike correlation, which is a measure of only linear dependency.

Since, Sklearn uses KNN to estimate MI; I first normalized all the features in the training set to be in the range [0, 1]. Further after computing MIs between individual features & response variable (fig_7), I selected all the features whose corresponding MI values > .001 (an arbitrary threshold chosen, keeping in mind the range of MI values). This brought down the dimensionality of feature space from 53 to 37, thereby reducing the curse of dimensionality in the process. In fact, the reduced feature dataset produced better test set roc_auc score for almost all the ML models employed in the study.

Furthermore, after feature selection, all the numerical features have been standardized both in the training & test datasets & the categorical variables represented by their dummy equivalents.

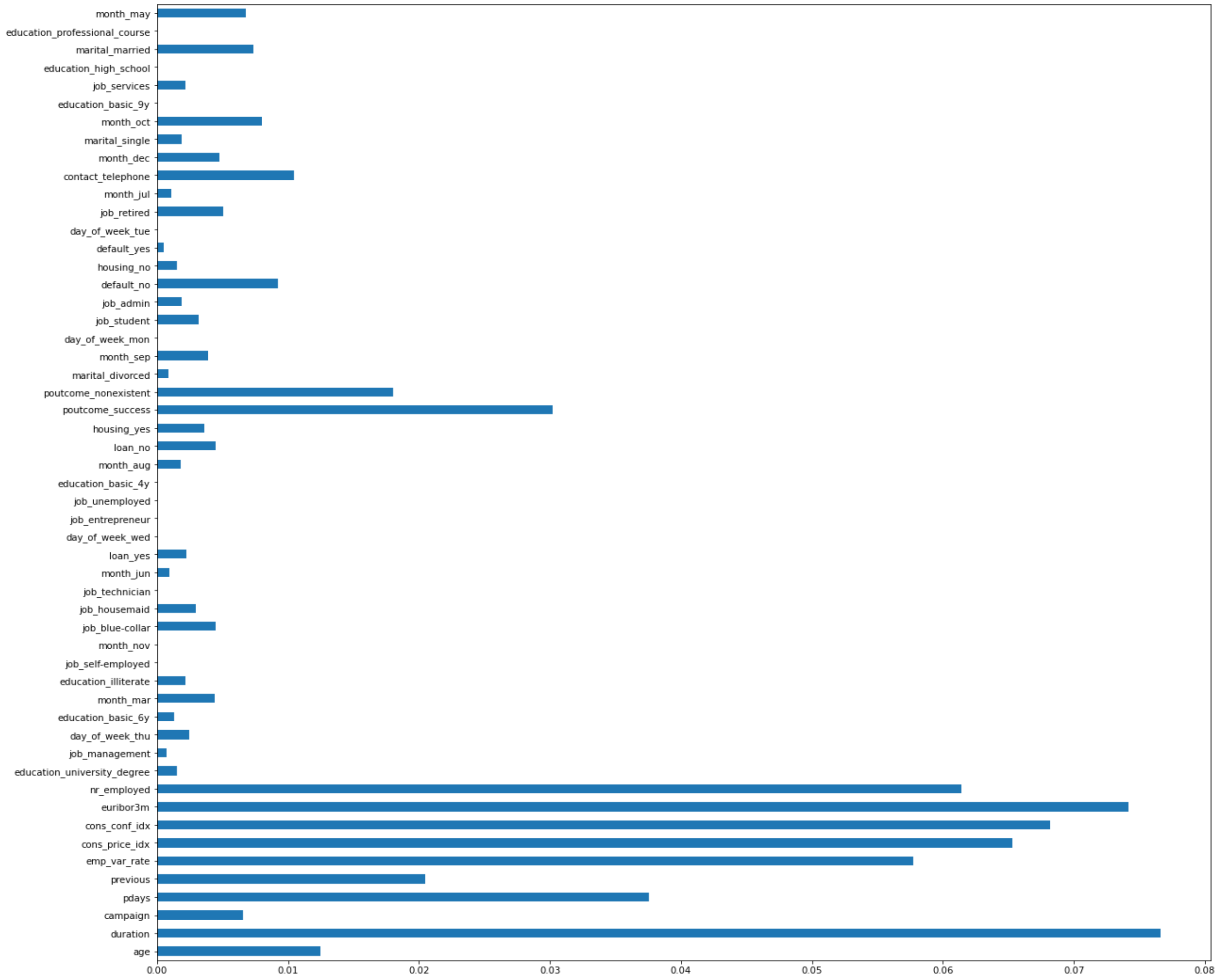


Fig_7: Mutual Information between Features & Response Variable (y).

Modeling

Baseline Model: Dummy Classifier with default parameters

Dummy classifier is a classifier that makes predictions using simple rules & is useful as a simple baseline to compare with other (real) classifiers.

The dummy classifiers with default parameters were fit on the full feature & reduced feature (reduced using mutual information) training set and the following observations are made:

a) The roc_auc score for the full feature training set using dummy classifier is 0.5013692527998519

b) The roc_auc score for the full feature test set using dummy classifier is 0.5050864722392566

c) The roc_auc score for the reduced feature training set using dummy classifier is 0.5013692527998519

d) The roc_auc score for the reduced feature test set using dummy classifier is 0.5050864722392566

e) The roc_auc score for the reduced feature test set is equal to that of full feature test set, indicating the presence of noise features in the full feature set.

f) The test set accuracies for both full feature & reduced feature test sets are less than 88.734%, which could be achieved by labeling all the test set instances (y) equal to 0, the % of class 0 (majority class) in the whole dataset. Thus the default dummy classifier is certainly underfitting the training set**.**

Model_1: Logistics Regression with tuned Hyper-parameters using Optuna.

Logistic Regression classifiers with tuned hyper-parameters (using model hyperparameter tuning library, Optuna) were fit on the full feature & reduced feature training set and the following observations are made:

a) The roc_auc score for the full feature training set using the tuned Logistic Regression is 0.7941151549648426

b) The roc_auc score for the full feature test set using the tuned Logistic Regression is 0.8009337852021322

c) The roc_auc score for the reduced feature training set using the tuned Logistic Regression is 0.7939385748220312

d) The roc_auc score for the reduced feature test set using the tuned Logistic Regression is 0.7986454254917684

e) Since the roc_auc score of the full feature test set using tuned Logistic regression model is almost equal to that of corresponding reduced feature test set, it confirms the earlier suspicion that there are lot of noise features in the original dataset.

f) The roc_auc test set score for the tuned Logistic models are much higher than the scores from the corresponding baseline models, which was expected.

g) The roc_auc test scores are more than their training counterparts for both full feature & reduced feature datasets, perhaps due to underfitting.

Defining Reward_Risk (R_R) Ratio for a Family of Machine Learning Models:

R_R Ratio = *Mean of K Fold CV Score Metric of Training Data / Std_Dev of K Fold CV Score Metric of Training Data*

The R_R ratio may be helpful in choosing among models having approximately same computational complexity or from the same family.

h) R_R Ratio for the tuned Logistic Regression using roc_auc metric is 49.17219276619884

Model_2: Random Forest Classifier with tuned Hyper-parameters using Optuna.

Random Forest Classifiers with tuned hyper-parameters (using model hyperparameter tuning library, Optuna) were fit on the full feature & reduced feature training set and the following observations are made:

a) The roc_auc score for the full feature training set using the tuned Random Forest classifier is 0.8813585851091872

b) The roc_auc score for the full feature test set using the tuned Random Forest classifier is 0.8142587061889712

c) The roc_auc score for the reduced feature training set using the tuned Random Forest classifier is 0.8575020591265264

d) The roc_auc score for the reduced feature test set using the tuned Random Forest classifier is 0.815116647601302

e) The above results clearly show that the full feature dataset contains noisy components. Thus we got better roc_auc for the reduced feature test set than full feature test set. Further the amount of over-fitting reduced when reduced feature test set was used. *Hence we would only use reduced feature set from now on.*

f) The R_R ratio for the tuned Random Forest Classifier (trained on reduced feature set) using roc_auc metric is: 52.904437509475876

g) The R_R Ratio as well as test set roc_auc for the tuned Random Forest classifier are more than those of corresponding Logistic Regression classifier, thus former fits the dataset better than the latter.

h) Random forest classifier over-fitting is generally due to high test set variance. Perhaps using extra trees classifier, which trades off high variance for higher bias, may help.

Model_3: Extra Trees Classifier with tuned Hyperparameters using Optuna.

Extra Trees Classifier with tuned hyper-parameters (using model hyperparameter tuning library, Optuna) was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Extra Trees classifier is 0.8384501171715599

b) The roc_auc score for the reduced feature test set using the tuned Extra Trees classifier is 0.8092702633378932

c) From the above analysis, we can clearly see that Extra Trees classifier did a good job in reducing over-fitting to a large extent, probably by reducing test set variance.

d) Though over-fitting was reduced, test set roc_auc score also suffered, probably due to corresponding increase in test set bias, which undermined any corresponding decrease in variance.

e) The R_R ratio for the best Extra trees Classifier using roc_auc metric is 52.1195604201456

f) ***Thus, keeping everything into account, for this dataset, the best Tree based bagging classifier is Optuna tuned Random Forest Classifier.***

Model_4: Linear Discriminant Analysis Classifiers

4.1) Vanilla LDA classifier with SVD solver

Vanilla LDA classifier (with no hyper-parameters) was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the Vanilla lda classifier, with SVD solver, is 0.7920662651019811

b) The roc_auc score for the reduced feature test set using the Vanilla lda classifier, with SVD solver, is 0.7966209343601113

c) The R_R ratio for the Vanilla lda classifier using roc_auc metric is: 51.152886235836306

4.2) LDA classifier with Eigen Solver with tuned Hyper-parameters using Optuna

LDA classifier, with Eigen Solver, & tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned LDA Classifier with Eigen solver is 0.7921019598225975

b) The roc_auc score for the reduced feature test set using the tuned LDA Classifier with Eigen solver is 0.7966579349261759

c) The R_R ratio for the tuned LDA classifier with Eigen solver using roc_auc metric is: 50.589549888479354

d) The tuned LDA model with Eigen solver as well as Vanilla LDA with SVD solver, performed worse than the Logistic Regression Model as well as Tree based Bagging models on the test set. This was expected as the underlying Feature space is not multivariate normal and doesn't have the same covariance matrix for both the classes, which are the underlying assumptions of the LDA models. However their R_R ratios were greater than that of the logistic Regression & less than those of Tree based Bagging Classifiers.

Model 5: Quadratic Discriminant Analysis Classifier (QDA) with tuned Hyper-parameters using Optuna

QDA classifier with tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned QDA classifier is 0.7851086555035252

b) The roc_auc score for the reduced feature test set using the tuned QDA classifier is 0.790439849757064

c) The R_R Ratio for the best QDA classifier using reduced feature set is 38.5216883550956

d) The tuned QDA model has the worst test set roc_auc score as well as R_R ratio of all the fitted models till now. This was expected as the underlying Feature space is not multivariate normal, which is an assumption of the QDA model. Any departure from normality affects the performance of QDA more than that of LDA, which is also observed here.

e) R_R Ratio for the best classifier in Discriminant Analysis family utilizing roc_auc metric is: 51.152886235836306, corresponding to Vanilla LDA Classifier.

f) ***Thus, keeping everything into account, for this dataset, the best discriminant classifier is LDA model with Eigen solver & tuned shrinkage.***

Model_6: Light-Gbm classifier with Tuned Hyperparameters using Optuna.

Light Gbm Classifier with tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Light Gbm classifier is 0.8252509438369647

b) The roc_auc score for the reduced feature test set using the tuned Light Gbm classifier is 0.817919919571678

c) The R_R Ratio for the tuned Light Gbm classifier, trained on reduced feature training set, using roc_auc score is 45.64661929591216

d) The tuned Light Gbm classifier has beautifully fitted the dataset with no indications of over-fitting and of all the classifiers tested till now, has given the best test set roc_auc score. May be Xg-Boost tested next may beat that.

e) Surprisingly R_R ratio for the tuned Light GBM model is on the lower side of the spectrum.

Model_7: XG-Boost Classifier with Tuned Hyperparameters using Optuna.

XG-Boost Classifier with tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned XgBoost classifier is 0.8384608606006355

b) The roc_auc score for the reduced feature test set using the best XgBoost classifier is 0.8171690734232746

c) The R_R Ratio for the best XgBoost classifier using roc_auc score is 48.43484617872931

d) XgBoost model, conceptually similar to Light GBM, wonderfully fitted this dataset with no apparent signs of over-fitting. But surprisingly the test set roc_auc score for tuned XgBoost is less than that of Light GBM.

e) XgBoost has a higher R_R ratio than that of Light GBM & hence should be preferred model of choice based on Rewards Risk ratio metric, but has a higher computational cost than that of Light Gbm.

f) Both boosting models have been outshone by Logistic Regression & Tree based Bagging models in the domain of R_R ratio metric, due to their comparatively low roc_auc score Std. Dev.

g) ***Thus keeping everything into account (especially the computational costs), for this dataset, the best Tree based boosting classifier is tuned Light Gbm classifier.***

Model_8: SVM Classifier with Tuned Hyperparameters using Random Search CV

SVM Classifier with 'rbf' kernel & tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned SVM classifier is 0.7853852343047899

b) The roc_auc score for the reduced feature test set using the tuned SVM classifier is 0.7934770655455445

c) The R_R Ratio for the best SVM classifier using roc_auc score is 47.16045814402432

d) SVM classifier shows no sign of over-fitting as both training and test roc_auc scores are almost equal to each other. But both training & test roc_auc scores for tuned SVM Model are less than the corresponding scores from the toned logistics Regression model (which belongs to the same classifier family as SVM).

e) SVM classifier fitted this dataset really really slowly & is only a feasible option for small or medium sized datasets.

f) **The R_R ratio for the tuned SVM model is less than that of best Logistic Regression, which has much lower computational complexity.** ***Thus for this dataset, Logistic Regression has outperformed SVM on all fronts.***

Model_9: Keras Variable Layer Dense Model with SELU Activation & Tuned LR Rate

Neural Net Classifier with SELU activation and tuned LR rate with Optuna tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Dense Neural Network classifier (with SELU activation & tuned LR rate) is 0.8134312507518557

b) The roc_auc score for the reduced feature test set using the tuned Dense Neural Network classifier (with SELU activation & tuned LR rate) is 0.8041722929147601

c) The R_R Ratio for the tuned variable layer Dense Neural Network (with variable no. of neurons/layer) classifier using reduced feature set is: 50.52183961321345

d) The Keras variable layer dense model with variable no. of neurons/layer did fit the training set well, with roc_auc score comparable to that of other models. But since Neural Networks work best when we have lots of data, we could have seen better results, with the above tuned model, if we had more data. May be tuned neural net with equal no. of neurons/ layer (evaluated next) offer better results.

e) The R_R ratio for the best Dense Neural Network (with variable no. of neurons/layer) is less than those of tree based bagging models, but more than those of tree based boosting models as well as Logistic Regression & SVM Models.

Model_10: Keras Variable Layer Dense Model with Equal No. of Neurons/layer, SELU Activation & Tuned learning rate.

Neural Net Classifier with Equal no. of Neurons/layer, SELU activation and tuned LR rate with Optuna tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Dense Neural Network (with equal no. of Neurons/layer) classifier is 0.8057215965111534

b) The roc_auc score for the reduced feature test set using the tuned Dense Neural Network (with equal no. of Neurons/layer) classifier is 0.805243687791877

c) The R_R Ratio for the best Dense Neural Network (with equal no. of neurons/layer) classifier using roc_auc metric is: 51.95977238913247

d) The Keras dense neural net (with equal no. of neurons/layer) fitted the Training set very well, with test set roc_auc score only less than those of tree based bagging & boosting models & beating the one associated with the previous more flexible neural network, while employing fewer layers & substantially less no. of tunable parameters.

e) The R_R ratio of the dense neural net (with equal no. of neurons/layer) is beaten only by those of Tree based bagging models & is quite high as compared to other classes of models*.* ***Hence after considering all the aspects, this model is the 2nd choice after Random Forest Classifier till now for this dataset.***

Model_11: Keras Variable Layer Dense Model with SELU Activation & Alpha_Dropout layer

Neural Net Classifier having variable no. of Neurons/layer, SELU activation & tuned LR rate & alpha_dropout layer with Optuna tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Dense Neural Network classifier (with SELU activation, tuned LR rate & alpha dropout layer) is 0.804127054187866

b) The roc_auc score for the reduced feature test set using the tuned Dense Neural Network classifier (with SELU activation, tuned LR rate & alpha dropout layer) is 0.8040792755554508

c) The R_R Ratio for the tuned variable layer Dense Neural Network (with variable no. of neurons/layer & alpha dropout layer) classifier using reduced feature set is: 52.04748626362193

d) The test set roc_auc score of the tuned Dense Neural Network (with Dropout) Classifier (even after employing considerably more no. of trainable parameters) is 0.8040792755554508, slightly less than that of with Neural net with equal no. of neurons/ layer. This was expected as regularization using dropout layer gives better result in dense neural nets where over-fitting is predominant, unlike shallow one here.

e) The R_R ratio of the tuned Dense Neural Network (with Dropout) Classifier is highest among the dense neural net family of classifier & comparable to that of tree based bagging classifiers, due to very small 10 fold roc_auc Std_Dev. This was expected as regularization techniques such as dropout tend to reduce test set variance.

f) Even though the R_R ratio for Best Neural Net with Dropout is comparable to those of Random Forest classifier, the former has high computational complexity & thus even here Random forest classifier is obvious winner.

Model_12: Keras Variable Layer Dense Model with SELU Activation & Alpha_Dropout layer, utilizing MC Alpha Dropout

Neural Net Classifier having variable no. of Neurons/layer, SELU activation & tuned LR rate & alpha_dropout layer, utilizing MC Alpha Dropout, with Optuna tuned hyper-parameters was fit on the reduced feature training set and the following observations are made.

a) The test set roc_auc score using the tuned MC _Dropout Keras Sequential classifier is: 0.8038853542619935

b) The test set roc_auc score for the Monte Carlo dropout model is slightly less than that of the alpha drop out model, indicating that the alpha drop out model fitted the dataset well.

c) ***Thus keeping everything into account (including No. of trainable parameters & the computational costs), for this dataset, the best Neural Net classifier is tuned Dense Network with equal no. of neurons/layer.***

Model_13: Voting Classifier with Soft Voting & Optuna tuned Weights.

Voting Classifier (with Soft Voting & Optuna tuned Weights) having component classifiers namely best tuned Logistic Regression, Random Forest, Light Gbm, LDA classifiers were fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Voting Classifier is 0.8327689992475545

b) The roc_auc score for the reduced feature test set using the tuned Voting Classifier is 0.8175044371196755

c) The R_R Ratio for the tuned Voting classifier using roc_auc metric is 47.28444783471113

d) The test set roc_auc score of the tuned voting classifier is more than that of all component models (by a good margin), but for that of Light GBM.

e) On the other hand, quite surprisingly the R_R ratio for the tuned voting classifier model is more than that of only Light GBM (among the component models).

f) Again after accounting for all the factors, the tuned Random forest classifier still reigns supreme.

Model_14: Voting Classifier with Tuned weights (without LDA component).

Voting Classifier (with Soft Voting, Optuna tuned Weights sans lda component) was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training set using the tuned Voting Classifier (without lda component) is 0.8340876676343492

b) The roc_auc score for the reduced feature test set using the tuned Voting Classifier (without lda component) is 0.8166479698334828

c) The R_R Ratio for the tuned Voting classifier (without lda) using roc_auc metric is 48.19861295727324

d) The voting classifier with lda outperformed the one without lda, in terms of test set roc_auc score, which was expected as the voting classifier performs well when the constituent models are many and well diversified.

e) The test set roc_auc score of the tuned voting classifier (without lda) is more than that of all component models (by a good margin), but for that of Light GBM.

f) Again, the R-R ratio for the tuned voting classifier (without lda) is more than that of only Light GBM (among the component models) & tuned voting classifier with lda.

g) ***Thus keeping everything into account (including the computational costs), for this dataset, the best Voting Classifier is the tuned Voting Classifier with lda. And the tuned Random forest classifier has overall beaten all the voting classifiers & is clearly the winner till here.***

Model_15: Blender Model

Selecting Random Forest Classifier as the Blender classifier, as it has one of highest test set roc_auc as well as R_R ratio for this dataset. Further Random Forest blender classifier (with Optuna tuned hyper-parameters) was fit on the reduced feature training set and the following observations are made.

a) The roc_auc score for the reduced feature training probabilities set using the tuned Random Forest Blender classifier is 0.8084145430361763

b) The roc_auc score for the reduced feature test probabilities set using the best Random Forest Blender classifier is 0.8130888396858342

c) The R_R Ratio for the Random Forest Blender Classifier using roc_auc metric is 27.396980488782663

d) The R_R ratio for the blender classifier is worst of all the classifiers, due to the high Std. Dev. of the CV roc_auc scores. This may be due to less training data being available to the blender classifier. ***Hence we can all, but rule out using blender classifier for this dataset.***

Model_16: Weighted Aggregating Classifier

This classifier computes the weighted aggregation of the predicted probabilities of constituent models (tuned Voting classifier & Neural net with equal neurons/layer here).Weighted Aggregating Classifier with Optuna tuned weights was fit on the reduced feature training set and the following observations are made.

a) The training set roc_auc score of the Weighted Aggregating Classifier using the Optuna tuned weights is 0.8004266845917924

b) The test set roc_auc score of the Weighted Aggregating Classifier using the Optuna tuned weights is 0.8173387453417614

c) The R_R Ratio for the tuned Weighted Aggregating Classifier using roc_auc metric is 47.57608261678961

d) The test set roc_auc score for the tuned Weighted Aggregating classifier is nearly equal to that of component tuned Voting Classifier, owing to more weight being assigned to it.

e) Similarly the R_R ratio of the Weighted Aggregating classifier is approx. equal (although more) to that of the component Voting Classifier. So with added complexity the former doesn't offer any advantage over the latter. ***Hence we can all, but rule out using weighted aggregating classifier for this dataset.***

Model_16: Stacking Classifier

Again selecting Random Forest Classifier as the Stacking classifier, as it has one of highest test set roc_auc, R_R ratio for this dataset as well as comparatively low computational cost. Furthermore Random Forest stacking classifier (with Optuna tuned hyper-parameters) was fit on the reduced feature training set and the following observations are made.

a) The test set roc_auc score for the tuned Random forest Stacking Classifier is: 0.8145040007783386

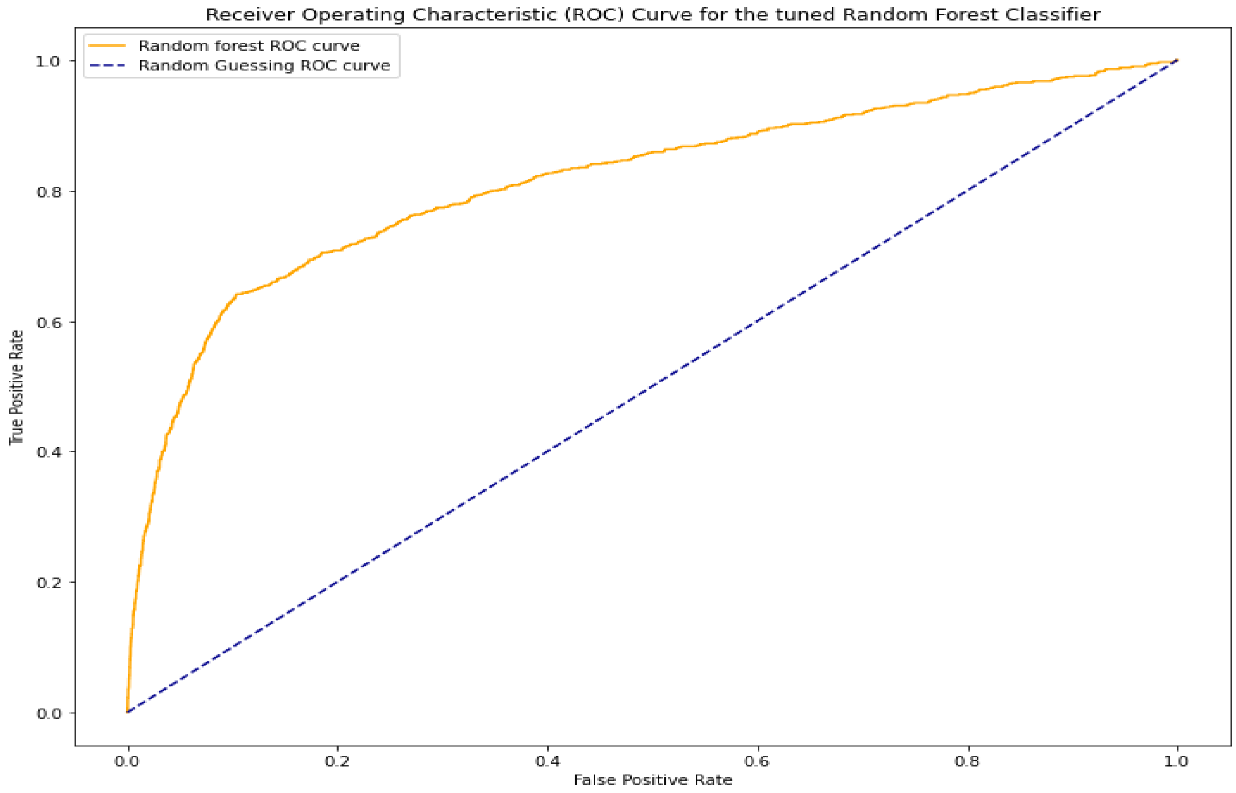
b) The R_R Ratio for the tuned Stacking Classifier using roc_auc metric is: 48.351649337461005

c) The test set roc_auc score for the Stacking classifier is more than that of component Neural Net classifier, but less than that of the component Voting classifier, both of which were used to create the training as well as test set for the stacking classifier.

d) Similarly the R_R ratio of the Stacking classifier is approx. equal (although more) to that of the component Voting Classifier, but much less than that of the Neural Net. Thus, even with added complexity, the Stacking Classifier still hasn't been able to beat the tuned Random Forest Classifier on this dataset. ***Hence we can all, but rule out using Stacking classifier for this dataset.***

Conclusion

Since the tuned Random Forest Classifier has one of the highest test set roc_auc score & Reward-Risk Ratio as well as one of the least training time complexity, it is the model of choice for this problem & dataset. Further let’s calculate the optimal probability threshold value from the roc curve that would give us the best TPR or Recall viz a viz FPR on the test or unseen data (*Note: Optimal Prob. threshold is one where diff between tpr & fpr is max.)*.



From the above roc plot, we have determined (fpr, tpr) point, with coordinates (0.1039671682626539, 0.6411637931034483,), corresponding to the optimal probability threshold (0.5186797366946175), which can be easily located on the graph. **Thus, in order to maximize recall viz. a viz. FPR, any test observation with conditional probability (outputted by RF classifier) > 0.5186797366946175, should be classified as belonging to class 1.**

Suggestions for further Improvements

Although, I have employed some feature selection and trained various models, some more feature engineering & training better ML models on this dataset can surely produce better R_R ratios as well as test set roc_auc scores. But this will require more business domain knowledge as well as computational resources especially when dealing with high computational cost models such as Neural Nets & SVMs. Making use of cloud computational resources can enable one to experiment with many different classifiers and their varied architectures and hence produce better results.