# Estimating Client Needs

## Business Case 2

T. Bucci, F. Cipriani, G. Corbo, D. Fabroni, M. Lucchini

Politecnico di Milano

March 31, 2022

# Table of Contents

# Goal: Recommendation System

We intend to estimate some investments needs for these customers using Data Science techniques.

**Data:**

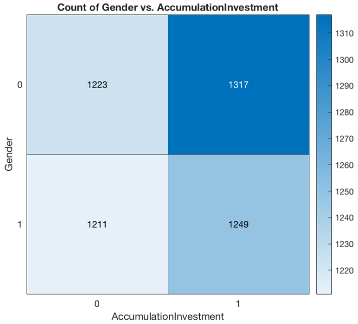
Needs:

`ID` `Age` `Gender` `FamilyMembers` `FinancialEducation` `RiskPropensity` `Income` `Wealth` `IncomeInvestment` `AccumulationInvestment`

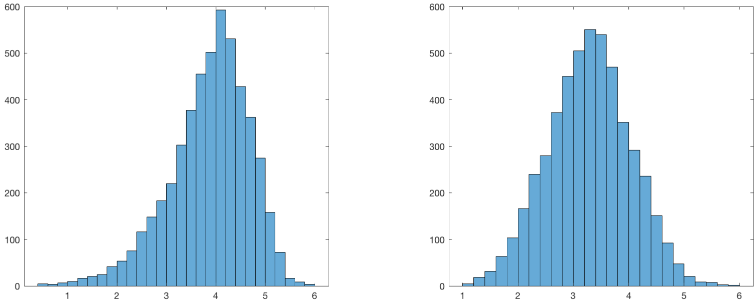
Products:

`IDProduct` `Type` `Risk`

# No significant sex difference




# Data Transformation

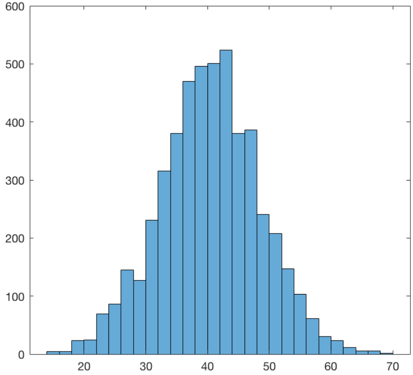



(a) log(Income) (b) Income$^{0.3}$

Figure: Comparison of transformations, we'll go with the second.

# Data Transformation

Using a boxcox transformation, $\text{NewAge} = (\text{OldAge}^{0.9} - 1)/0.9$.



# Data Transformation

$$\text{NewFinancialEducation} = \text{OldFinancialEducation}^{0.8}$$
$$\text{NewRiskPropensity} = \text{OldRiskPropensity}^{0.65}$$
$$\text{NewIncome} = \text{OldIncome}^{0.3}$$

# Cross validation

```
nObs = size(Data, 1);
rng(10)
idxPermutation = randperm(nObs);
X = X(idxPermutation,:); % random permutation
train =  0.75  ;
cross =     0  ;

nObsTrain = round(train*nObs);
nObsCross = round(cross*nObs);

XTrain = X(1:nObsTrain,:);
XCross = X((nObsTrain+1):(nObsTrain+nObsCross), :);
XTest = X(nObsTrain+1+nObsCross:end,:);
```

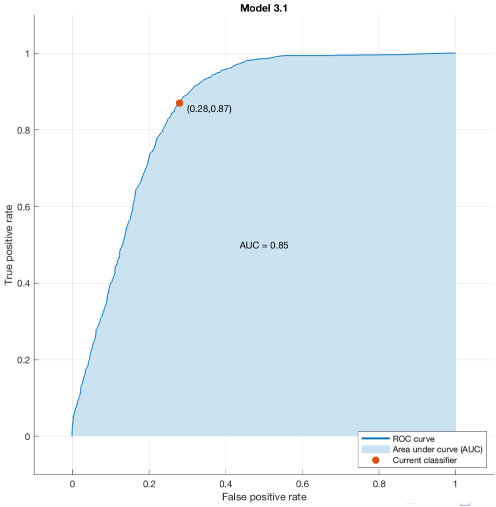
# Added $Y$ permutation equal to $X$

```
YInc = Data.IncomeInvestment;
YInc = YInc(idxPermutation);

yInvIncTrain = YInc(1:nObsTrain);
yInvIncCross = YInc((nObsTrain+1):(nObsTrain+nObsCross));
yInvIncTest = YInc(nObsTrain+nObsCross+1:end);

YAcc = Data.AccumulationInvestment;
YAcc = YAcc(idxPermutation);

yInvAccTrain = YAcc(1:nObsTrain);
yInvAccCross = YAcc((nObsTrain+1):(nObsTrain+nObsCross));
yInvAccTest = YAcc(nObsTrain+nObsCross+1:end);

varNames = {'Age', 'Gender', 'Family', 'FinEdu', 'Risk', 'Income', 'Wealth'};
XTrainTable=table(XTrain(:,1), XTrain(:,2), XTrain(:,3), XTrain(:,4),
XTrain(:,5), XTrain(:,6),XTrain(:,7), 'VariableNames',varNames);
XTestTable=table(XTest(:,1), XTest(:,2), XTest(:,3), XTest(:,4),
XTest(:,5), XTest(:,6),XTest(:,7), 'VariableNames',varNames);
```

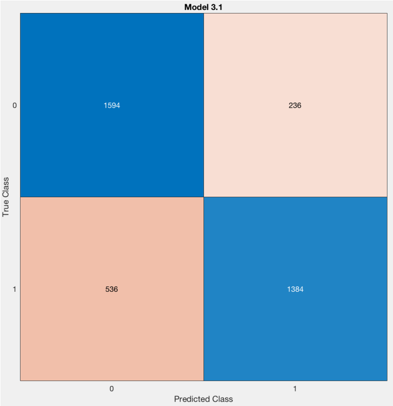
# Classification Learner: model training

| Model | Type | Last change | Accuracy | Features |
|---|---|---|---|---|
| 1.4 | KNN | Fine KNN | 65.7% | 7/7 features |
| 1.5 | KNN | Medium KNN | 69.8% | 7/7 features |
| 1.6 | KNN | Coarse KNN | 68.9% | 7/7 features |
| 1.7 | KNN | Cosine KNN | 69.8% | 7/7 features |
| 1.8 | KNN | Cubic KNN | 69.1% | 7/7 features |
| 1.9 | KNN | Weighted KNN | 69.7% | 7/7 features |
| 2 | Ensemble | Boosted Trees | **79.4%** | 7/7 features |
| 3.1 | Ensemble | Boosted Trees | **79.4%** | 7/7 features |
| 3.2 | Ensemble | Bagged Trees | 76.6% | 7/7 features |
| 3.3 | Ensemble | Subspace Discrimin... | 62.9% | 7/7 features |
| 3.4 | Ensemble | Subspace KNN | 70.0% | 7/7 features |
| 3.5 | Ensemble | RUSBoosted Trees | 75.0% | 7/7 features |

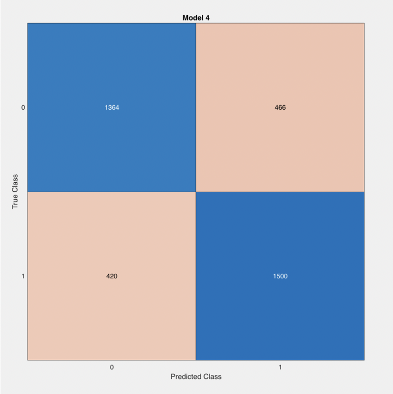
# Classification Learner: Boosted Tree ROC



# Classification Learner: Boosted Tree confusion matrix




(a) Confusion matrix.




(b) Penalized confusion matrix.

# Reducing data

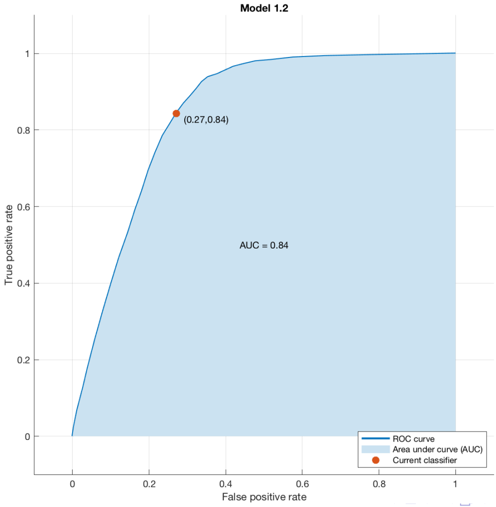
We remove `Gender`, `Risk` and `FamilySize` and added `IncomeWealthRatio` = `Income` / `Wealth`.

```
Xsmall = [rescale(IncomeWealthRatio) rescale(pAge) rescale(pFin) rescale(pIncome) rescale(pWealth)];
Xsmall = Xsmall(idxPermutation,:);
XsmallTrain = Xsmall(1:nObsTrain,:);
XsmallTest = Xsmall(nObsTrain+1:end,:);
XsmallTrainTable=table(XsmallTrain(:,1), XsmallTrain(:,2), XsmallTrain(:,3),
XsmallTrain(:,4), XsmallTrain(:,5), 'VariableNames',xnames);
```

# Classification Learner: model training

| | | | |
|---|---|---|---|
| 1.1 | Ensemble | Last change: Boosted Trees | Accuracy: 78.3% 5/5 features |
| 1.2 | Ensemble | Last change: Bagged Trees | $\boxed{\text{Accuracy: } \mathbf{78.5\%}}$ 5/5 features |
| 1.3 | Ensemble | Last change: Subspace Discrimin... | Accuracy: 65.9% 5/5 features |
| 1.4 | Ensemble | Last change: Subspace KNN | Accuracy: 73.4% 5/5 features |
| 1.5 | Ensemble | Last change: RUSBoosted Trees | Accuracy: 76.3% 5/5 features |
| 2.1 | Tree | Last change: Fine Tree | Accuracy: 78.0% 5/5 features |
| 2.2 | Tree | Last change: Medium Tree | Accuracy: 76.1% 5/5 features |
| 2.3 | Tree | Last change: Coarse Tree | Accuracy: 68.0% 5/5 features |
| 2.4 | KNN | Last change: Fine KNN | Accuracy: 70.5% 5/5 features |

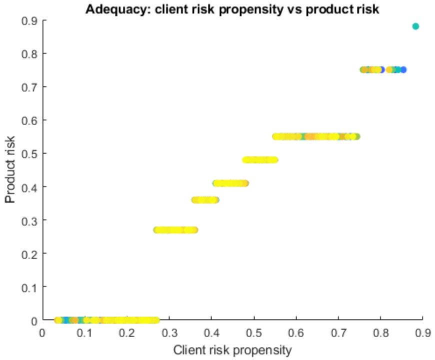
# Classification Learner: Bagged Tree ROC



# Recommending products

Using MATLAB's automatic hyperparameters optimization we decide to use the Bagged Tree model after evaluating 30 alternatives.

# Product recommendation based on Risk Propensity



- more data cleaning to improve model accuracy;
- better data visualization for product recommendation.