1. Logistic

[[3528 17]

[ 211 24]]

Accuracy: 0.9396825396825397

1. Random Forest (max\_features = sqrt)

[[3515 30]

[ 81 154]]

precision recall f1-score support

0 0.98 0.99 0.98 3545

1 0.84 0.66 0.74 235

accuracy 0.97 3780

macro avg 0.91 0.82 0.86 3780

weighted avg 0.97 0.97 0.97 3780

0.9706349206349206

1. Random Forest (max\_features = log2)

[[3516 29]

[ 93 142]]

precision recall f1-score support

0 0.97 0.99 0.98 3545

1 0.83 0.60 0.70 235

accuracy 0.97 3780

macro avg 0.90 0.80 0.84 3780

weighted avg 0.97 0.97 0.97 3780

0.9677248677248678

1. Feature importance

Variable: Ratio\_crdt\_trnovr\_0\_6mois Importance: 0.08

Variable: Ratio\_dbt\_bal\_0\_12mois Importance: 0.08

Variable: top\_clot\_cav12 Importance: 0.07

Variable: Ratio\_dbt\_bal\_0\_6mois Importance: 0.07

Variable: Ratio\_dbt\_trnovr\_0\_12mois Importance: 0.05

Variable: Ratio\_dbt\_trnovr\_0\_6mois Importance: 0.04

Variable: change\_nb\_prod Importance: 0.04

Variable: Avg\_Crdt\_trnovr\_6mois Importance: 0.03

Variable: Ratio\_crdt\_trnovr\_6\_12mois Importance: 0.03

Variable: Ratio\_dbt\_bal\_6\_12mois Importance: 0.03

Variable: MONS\_INLIM\_NB\_PROD Importance: 0.03

Variable: MONS\_INLIM\_SERCL Importance: 0.03

Variable: Avg\_Dbt\_Bal\_6mois Importance: 0.02

Variable: Avg\_Dbt\_Bal\_12mois Importance: 0.02

Variable: Avg\_Crdt\_trnovr\_12mois Importance: 0.02

Variable: Avg\_Dbt\_trnovr\_6mois Importance: 0.02

Variable: Avg\_Dbt\_trnovr\_12mois Importance: 0.02

Variable: Avg\_Savngs\_Bal\_12mois Importance: 0.02

Variable: Ratio\_dbt\_trnovr\_6\_12mois Importance: 0.02

Variable: Ratio\_Savngs\_bal\_0\_6mois Importance: 0.02

Variable: Ratio\_Savngs\_bal\_0\_12mois Importance: 0.02

Variable: Ratio\_Savngs\_bal\_6\_12mois Importance: 0.02

Variable: mt\_rvnu Importance: 0.02

Variable: Change\_TRB Importance: 0.02

Variable: Change\_cr Importance: 0.02

Variable: Change\_db Importance: 0.02

Variable: MOB(yrs) Importance: 0.01

Variable: age Importance: 0.01

Variable: Avg\_Savngs\_Bal\_6mois Importance: 0.01

Variable: MONS\_INLIM\_TRB Importance: 0.01

Variable: MONS\_INLIM\_CARS Importance: 0.01

Variable: MONS\_INLIM\_DAV Importance: 0.01

Variable: MONS\_INLIM\_CRFLOW Importance: 0.01

Variable: MONS\_INLIM\_DRFLOW Importance: 0.01

Variable: MONS\_INLIM\_PROCL Importance: 0.01

**ACCOUNT FOR 95% IMPORTANCE**

Variable: TOP\_PERSO Importance: 0.0

Variable: TOP\_IMMO Importance: 0.0

Variable: TOP\_CAV Importance: 0.0

Variable: TOP\_GSM Importance: 0.0

Variable: TOP\_ASSV Importance: 0.0

Variable: TOP\_CB\_VISA Importance: 0.0

Variable: TOP\_CB\_BUSI Importance: 0.0

Variable: TOP\_CB\_PREMIER Importance: 0.0

Variable: TOP\_CB\_INFI Importance: 0.0

Variable: TOP\_CB\_MASTER Importance: 0.0

Variable: TOP\_CONV\_M\_1 Importance: 0.0

Variable: top\_clot\_pea12 Importance: 0.0

Variable: top\_clot\_IMMO12 Importance: 0.0

Variable: top\_clot\_PERSO12 Importance: 0.0

Variable: top\_staff Importance: 0.0

Variable: TOP\_PEA Importance: 0.0

Variable: TOP\_TITRE Importance: 0.0

Variable: TOP\_INV Importance: 0.0

Variable: top\_cars Importance: 0.0

Variable: top\_carte Importance: 0.0

Variable: TOP\_PAssport Importance: 0.0

Variable: Contact\_3mois Importance: 0.0

Variable: Contact\_6mois Importance: 0.0

Variable: Contact\_9mois Importance: 0.0

Variable: Contact\_12mois Importance: 0.0

Variable: Avg\_Wealth\_Bal\_6mois Importance: 0.0

Variable: Avg\_Wealth\_Bal\_12mois Importance: 0.0

Variable: NB\_PE\_ME Importance: 0.0

Variable: mt\_dbloc\_cons\_rep Importance: 0.0

Variable: mt\_capir\_cons\_rep Importance: 0.0

Variable: mt\_dbloc\_immo\_rep Importance: 0.0

Variable: mt\_capir\_immo\_rep Importance: 0.0

Variable: MONS\_INLIM\_ASV Importance: 0.0

Variable: MONS\_INLIM\_PORT Importance: 0.0

1. Gradient boosting

learning\_rate=0.005, n\_estimators=100,max\_depth=9, min\_samples\_split=1200, min\_samples\_leaf=60, random\_state=42, max\_features=7,

warm\_start=True

Model Report

Accuracy : 0.9364

AUC Score (Train): 0.973177

Model Report Test

Accuracy : 0.9378

AUC Score (Test): 0.970488

1. XG Boost

learning\_rate =0.005, n\_estimators=100, max\_depth=5, min\_child\_weight=1, gamma=0, subsample=0.8, colsample\_bytree=0.8, objective= 'binary:logistic', nthread=4, scale\_pos\_weight=1, seed=42

Model Report

Accuracy : 0.9724

AUC Score (Train): 0.981456

Model Report Test

Accuracy : 0.9672

AUC Score (Test): 0.976455

1. ANN

hidden\_layer\_sizes=(4,4,4), activation=**'relu'**, solver='adam',max\_iter=1000,learning\_rate\_init=0.01

[[5251 74]

[ 150 195]]

precision recall f1-score support

0 0.97 0.99 0.98 5325

1 0.72 0.57 0.64 345

accuracy 0.96 5670

macro avg 0.85 0.78 0.81 5670

weighted avg 0.96 0.96 0.96 5670

0.9604938271604938

hidden\_layer\_sizes=(4,4,4), activation=**'logistic'**, solver='adam',max\_iter=1000,learning\_rate\_init=0.01

[[5250 75]

[ 123 222]]

precision recall f1-score support

0 0.98 0.99 0.98 5325

1 0.75 0.64 0.69 345

accuracy 0.97 5670

macro avg 0.86 0.81 0.84 5670

weighted avg 0.96 0.97 0.96 5670

0.9650793650793651

hidden\_layer\_sizes=(4,4,4), activation=**'tanh'**, solver='adam',max\_iter=1000,learning\_rate\_init=0.01

[[5208 117]

[ 104 241]]

precision recall f1-score support

0 0.98 0.98 0.98 5325

1 0.67 0.70 0.69 345

accuracy 0.96 5670

macro avg 0.83 0.84 0.83 5670

weighted avg 0.96 0.96 0.96 5670

0.9610229276895944

hidden\_layer\_sizes=(4,4,4), activation=**'identity'**, solver='adam',max\_iter=1000,learning\_rate\_init=0.01

[[5275 50]

[ 169 176]]

precision recall f1-score support

0 0.97 0.99 0.98 5325

1 0.78 0.51 0.62 345

accuracy 0.96 5670

macro avg 0.87 0.75 0.80 5670

weighted avg 0.96 0.96 0.96 5670

0.9613756613756613

trying for sgd etc with logistic as it performed best

hidden\_layer\_sizes=(4,4,4), activation='logistic', solver='**lbfgs**', max\_iter=1000,learning\_rate\_init=0.01

[[5254 71]

[ 143 202]]

precision recall f1-score support

0 0.97 0.99 0.98 5325

1 0.74 0.59 0.65 345

accuracy 0.96 5670

macro avg 0.86 0.79 0.82 5670

weighted avg 0.96 0.96 0.96 5670

0.962257495590829

hidden\_layer\_sizes=(4,4,4), activation='logistic', solver='**adam**',max\_iter=1000,learning\_rate\_init=0.01

[[5250 75]

[ 123 222]]

precision recall f1-score support

0 0.98 0.99 0.98 5325

1 0.75 0.64 0.69 345

accuracy 0.97 5670

macro avg 0.86 0.81 0.84 5670

weighted avg 0.96 0.97 0.96 5670

0.9650793650793651

hidden\_layer\_sizes=(4,4,4), activation='logistic', solver='**sgd**',max\_iter=1000,learning\_rate\_init=0.01

[[5258 67]

[ 152 193]]

precision recall f1-score support

0 0.97 0.99 0.98 5325

1 0.74 0.56 0.64 345

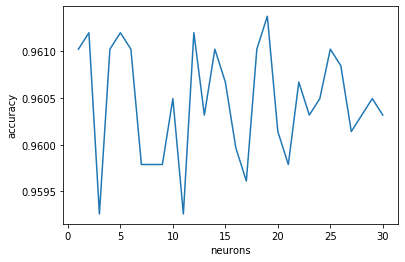
accuracy 0.96 5670

macro avg 0.86 0.77 0.81 5670

weighted avg 0.96 0.96 0.96 5670

0.9613756613756613

FOR LOOP FOR CHECKING with respect to the number of neurons



ANN using tensorflow

model = tf.keras.models.Sequential([

*# 512 neuron hidden layer*

tf.keras.layers.Dense(70, activation='relu'),

tf.keras.layers.Dense(32, activation='relu',kernel\_regularizer= tf.keras.regularizers.l2(l=0.2)),

*#tf.keras.layers.Dense(24, activation='relu'),*

*#tf.keras.layers.Dense(16, activation='relu'),*

tf.keras.layers.Dense(8, activation='relu'),*# kernel\_regularizer= tf.keras.regularizers.l2(l=0.2)),*

*# Only 1 output neuron. It will contain a value from 0-1 where 0 for 1 class ('horses') and 1 for the other ('humans')*

tf.keras.layers.Dense(1, activation='sigmoid')

])

**from** **tensorflow.keras.optimizers** **import** RMSprop

model.compile(loss='binary\_crossentropy',

*#optimizer = tf.keras.optimizers.SGD(lr=0.1, momentum=0.9, decay=0.01),*

*#optimizer = RMSprop(lr=0.1),*

optimizer = 'adam',

*#optimizer = tf.keras.optimizers.SGD(lr=0.1),*

metrics=['acc'])

