

Mitra UMI Assessment Scoring Model

Tim Ultra Mikro BRI

This document and information contained herein are confidential and proprietary to PT. Bank Rakyat Indonesia, Tbk and shall not be published or disclosed to any third party without the express written consent by an authorized representative of PT. Bank Rakyat Indonesia, Tbk



Sebagai salah satu proses pemilihan mitra umi untuk bekerja sama dengan BRI



Dapat mendapatkan mitra umi yang produktif baik dari sisi pencapaian maupun portofolionya



Dapat mengurangi terjadinya revenue loss akibat mitra umi yang tidak produktif

1



Transasaksi

Merupakan histori fitur transaksi yang dilakukan sebagai agen brilink

2



Performa Referral

Merupakan kolektibilitas nasabah yang pernah direferalkan

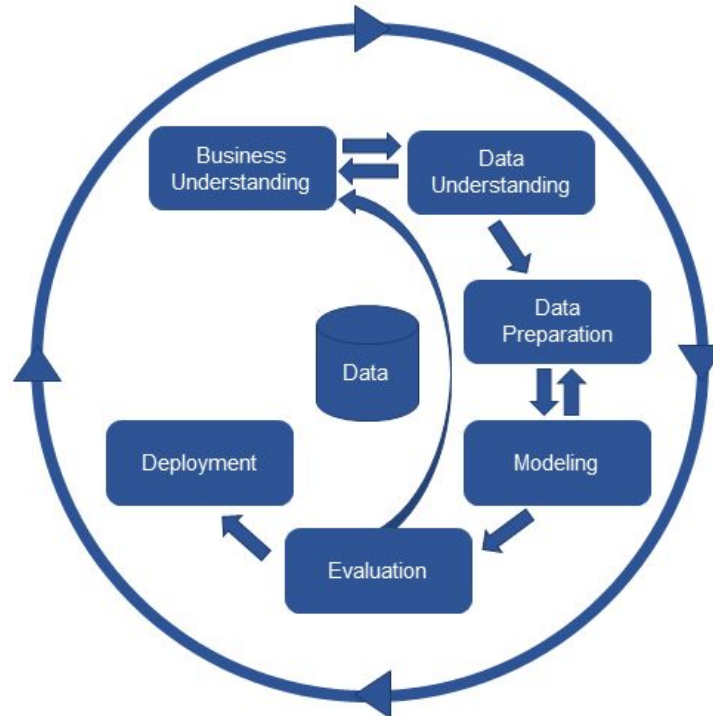
3



Demografi

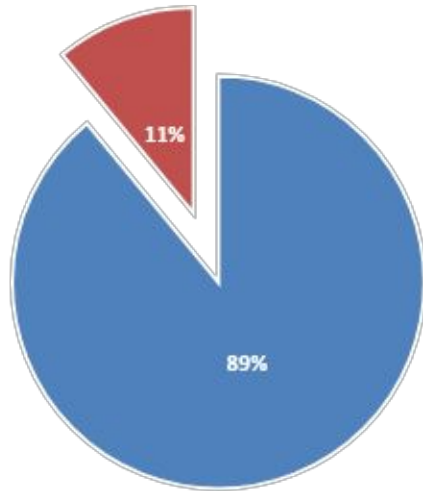
Merupakan demografi dan data – data terkait individu calon mitra umi

CRISP – DM CRoss Industry Standard Processing for Data Mining



Data Source Identification

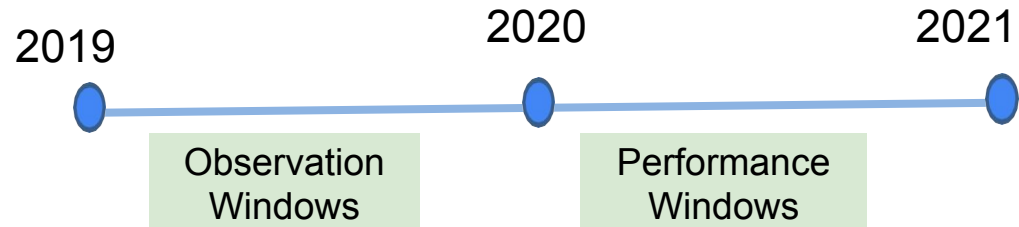
Proporsi Label



Vintage Analysis



Grafik yang menunjukkan perbandingan dari kolektibilitas terakhir rekening yang direferalkan dengan kolektibilitas maksimum pada rekening pinjaman



Observasi performa pinjaman yang direferlakan diamati dari 2019 s/d 2020. sedangkan performa agen brilink diamati dari 2020 s/d 2021.

Summary Exploratory Data Analysis - Historis Transaksi



Setoran Simpanan

Frekuensi transaksi setoran simpanan oleh mitra rata – rata sebanyak 3 kali per bulan



Setoran Pinjaman

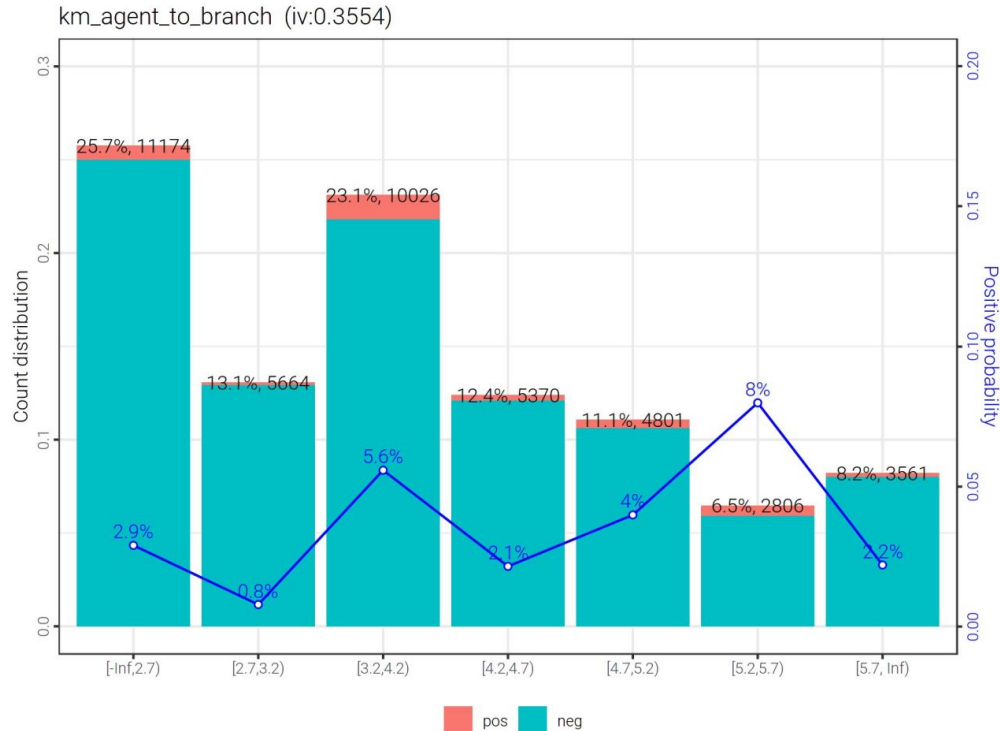
Frekuensi transaksi setoran pinjaman oleh mitra rata – rata sebanyak 1 kali per bulan



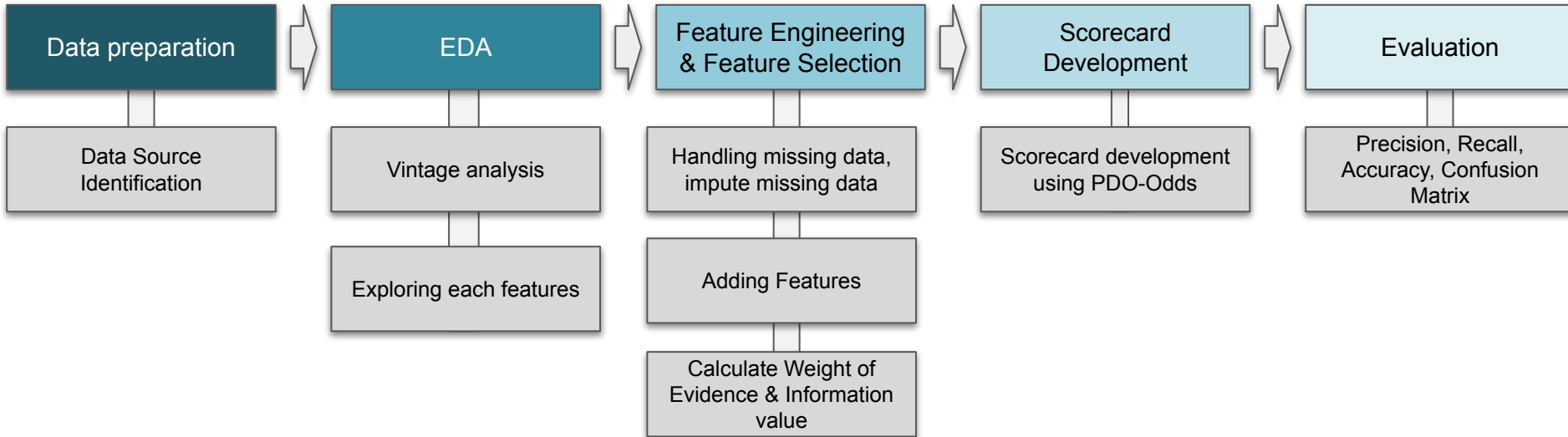
Referral Pinjaman

Rata – rata tenor yang direferalkan sekitar 3,3 juta

Summary Exploratory Data Analysis - Kondisi Kantor Cabang



semakin jauh jarak mitra dari unit kerja terlihat semakin besar kemungkinan nasabah referral yang menunggak.



Top 10 Feature yang paling Signifikan

No	Feature	Information Value
1	amt_setoranpinjaman_med (Nominal Setoran Pinjaman)	0.343517
2	freq_setoranpinjaman_med (Frekuensi Setoran Pinjaman)	0.233471
3	freq_tariktunai_med (Frekuensi Tarik Tunai)	0.218918
4	amt_setoransimpanan_med (Nominal Setoran Simpanan)	0.205119
5	freq_pembelian_med (Frekuensi Transaksi Pembelian)	0.160754
6	amt_pembelian_med (Nominal Transaksi Pembelian)	0.149968
7	freq_setoransimpanan_med (Frekuensi Setoran Simpanan)	0.1424
8	freq_pembayaran_med (Frekuensi Transaksi Pembayaran)	0.135929
9	amt_pembayaran_med (Nominal Transaksi Pembayaran)	0.107165
10	amt_tariktunai_med (Nominal Transaksi Tarik Tunai)	0.095151

Strong Prediction



Weak Prediction

create function

```
IV_function <- function(x, y=df$Status){  
  mt <- as.matrix(table(as.factor(x), as.factor(y)))  
  Total <- mt[,1] + mt[,2]  
  Total_Pct <- round(Total/sum(mt)*100, 2)  
  Bad_pct <- round((mt[,1]/sum(mt[,1]))*100, 2)  
  Good_pct <- round((mt[,2]/sum(mt[,2]))*100, 2)  
  Bad_Rate <- round((mt[,1]/(mt[,1]+mt[,2]))*100, 2)  
  grp_score <- round((Good_pct/(Good_pct + Bad_pct))*10, 2)  
  WOE <- round(log(Good_pct/Bad_pct)*10, 2)  
  g_b_comp <- ifelse(mt[,1] == mt[,2], 0, 1)  
  IV <- ifelse(g_b_comp == 0, 0, (Good_pct - Bad_pct)*(WOE/10))  
  Efficiency <- abs(Good_pct - Bad_pct)/2  
  otb<-as.data.frame(cbind(mt, Good_pct, Bad_pct, Total,  
                           Total_Pct, Bad_Rate, grp_score,  
                           WOE, IV, Efficiency ))  
  
  otb$Names <- rownames(otb)  
  rownames(otb) <- NULL  
  otb[,c(12,2,1,3:11)]  
}
```

Using Library

```
library('Information')  
df$Status <- as.factor(df_train_woe$Status)  
IV <- InformationValue::IV(X=df, Y=df$Status)  
  
variable <- data.frame(IV$Summary[round(IV$Summary$IV,2)>0.02 & round(IV$Summary$IV,2)<1,])  
  
df <- df[,names(df)  
      %in% c(variable$Variable,"Status")]
```

menghasilkan

6

Pohon Keputusan

```
xgb_learner <- makeLearner(
  "classif.xgboost",
  predict.type = "response",
  par.vals = list(
    objective = "binary:logistic",
    eval.metric = "auc",
    nrounds = 2000
  )
)

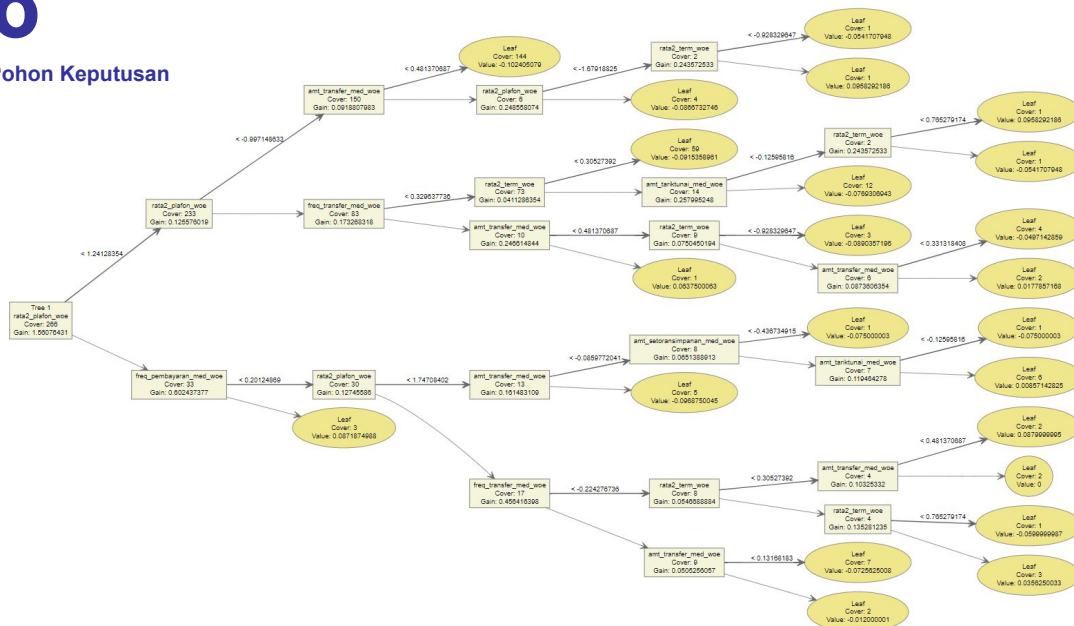
xgb_params <- makeParamSet(
  makeIntegerParam("nrounds", lower = 100, upper = 500),
  makeDiscreteParam("booster", values = c("gbtree")),
  makeNumericParam("min_child_weight", lower = 1L, upper = 10L),
  makeNumericParam("subsample", lower = 0.5, upper = 1),
  makeNumericParam("colsample_bytree", lower = 0.5, upper = 1),
  makeIntegerParam("max_depth", lower = 1, upper = 10),
  makeNumericParam("eta", lower = .1, upper = .5),
  makeNumericParam("lambda", lower = -.1, upper = 0, trafo = function(x) 10^x)
)

control <- makeControlRandom(maxit = 5L)
resample_desc <- makeResampleDesc("CV", iters = 4L, stratify = T)
tuned_params <- tuneParams(
  learner = xgb_learner,
  task = trainTask,
  resampling = resample_desc,
  par.set = xgb_params,
  control = control
)

xgb_tuned_learner <- setHyperPars(
  learner = xgb_learner,
  par.vals = tuned_params$ix
)

xgb.train.data = xgb.DMatrix(data.matrix(df_train_data), label = as.numeric(as.character(df_train_label)), missing = NA)
xgb.test.data = xgb.DMatrix(data.matrix(df_test_data), label = as.numeric(as.character(df_test_label)), missing = NA)

param <- xgb_tuned_learner$par.vals
set.seed(1)
cv <- createfolds(df_train_label, k = 4L)
xgboost.cv <- xgb.cv(param=param, data = xgb.train.data, folds = cv, early_stopping_rounds = 5, nrounds = 500)
best_iteration <- xgboost.cv$best_iteration
xgb.model <- xgboost(param=param, data = xgb.train.data, nrounds=best_iteration, eval = "auc")
```



Actual Values	Predicted Values	
	Predicted: Good (-)	Predicted: Bad (+)
	Actual: Good (-)	Actual: Bad (+)
	TN (1788)	FP (148)
	FN (135)	TP (1801)

Evaluation	Rumus	Hasil
Accuracy	$\frac{TP + TN}{(TP + TN + FP + FN)}$	94%
Precision	$\frac{TP}{TP + FP}$	92.4%
Recall (Sensitivity)	$\frac{TP}{TP + FN}$	93.02%
F1 Score	$\frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$	92.71%
AUC	Area Under ROC Curve TP dan FP	96%

Penjelasan:

- True Positive (TP) = Nasabah diprediksi **bad** dan aktualnya memang **bad**
- True Negative (TN) = Nasabah diprediksi **good** dan aktualnya memang **good**
- False Positive (FP) = Nasabah diprediksi **bad** namun aktualnya **good**
- False Negative (FN) = Nasabah diprediksi **good** namun aktualnya **bad**

- Accuracy = model bisa memprediksi yang benar dan yang salah
- Precision = model bisa memprediksi yang benar bad dari keseluruhan prediksi bad contoh lain: memprediksi kanker, lebih aman hasil prediksi banyak yang kanker karena jika yang aktualnya kanker tidak diprediksi kanker, taruhannya nyawa
- Recall = model bisa memprediksi yang benar bad dari keseluruhan aktual bad contoh lain: memprediksi email spam, lebih aman hasil prediksi tidak spam karena bisa jadi ada email pembayaran yg penting malah dianggap spam/
- F1 Score = rata-rata dari nilai presisi dan recall
- AUC = ukuran kemampuan model dalam mengenali nasabah bad dan tidak bad

Before



Penilai kelayakan mitra
umi berdasarkan
assessment dari mantri



*Terdapat Mitra UMI yang tidak
produktif padahal menurut
penilaian mantri ybs berpotensi*

After



Setiap mitra umi memiliki
skor yang menunjukkan
potensi dari calon mitra
umi



Kantor pusat memiliki
engine untuk melakukan
penilaian terhadap mitra
umi

Dengan menggunakan
model prediksi, dapat
mengoptimalkan proses
bisnis, pemilihan mitra umi
dapat dilakukan dengan
akurat dan cepat

Thank You

This document and information contained herein are confidential and proprietary to PT. Bank Rakyat Indonesia, tbk and shall not be published or disclosed to any third party without the express written consent by an authorized representative of PT. Bank Rakyat Indonesia, tbk