

Nama : Vina  
Jurusan : Ekonomi Pembangunan  
Asal Kampus : Universitas Sriwijaya  
Program : Accelerated Machine Learning

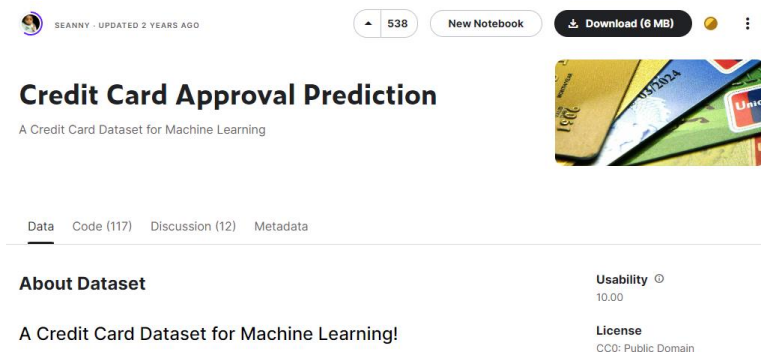
---

### ***CREDIT SCORING IN BANKING***

*Credit scoring* adalah sistem yang digunakan oleh pihak bank untuk menentukan apakah nasabah layak untuk diberikan pinjaman atau tidak. Dalam menentukan *credit scoring*, data profil nasabah sangat diperlukan. Semakin lengkap dan tepat data nasabah yang tersedia, maka semakin akurat *credit scoring*-nya. Di era serba digital ini membuat nasabah memiliki banyak kemudahan dalam meminjam uang sehingga diperlukan *machine learning* untuk mengolah data profil nasabah dengan efisien. Hal ini bertujuan agar dapat bersaing dengan bank lainnya, tetapi tetap mendapatkan peminjam yang memiliki tingkat *credit scoring* yang tinggi. Melalui *machine learning* ini memudahkan pihak bank untuk mengetahui mana saja nasabah-nasabah yang layak untuk diberi pinjaman.

Dalam credit scoring, terdapat empat tahapan. Pada tahapan pertama adalah dataset. Dataset merupakan kumpulan data. Pada *credit scoring in banking*, dataset yang akan digunakan adalah data credit card profil yang diperoleh dari *credit card approval prediction* dari kaggle.

**Gambar 1. *Credit Card Approval Prediction***



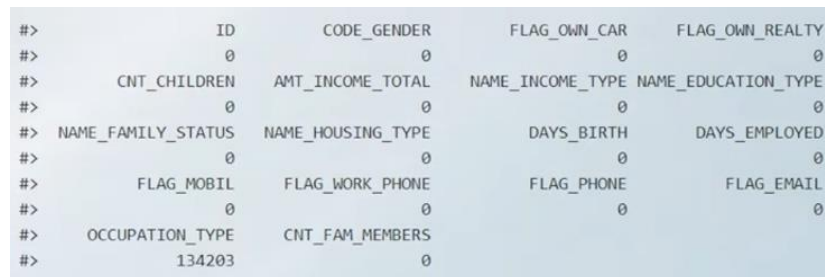
Sumber: kaggle

Adapun variabel yang digunakan meliputi jenis kelamin, pendapatan tahunan, jumlah anak jika sudah berkeluarga, tingkat pendidikan, ataupun status pernikahan. Dari tahapan ini akan dilanjutkan dengan memprediksi apakah nasabah tersebut merupakan nasabah layak atau tidak untuk diberikan pinjaman.

Setelah dataset, tahapan kedua adalah cleaning data. Cleaning data adalah prosedur untuk memastikan apakah data tersebut benar dan konsisten dalam dataset. Ada empat tahapan dalam cleaning data, yaitu:

1. Cek *Missing value*.

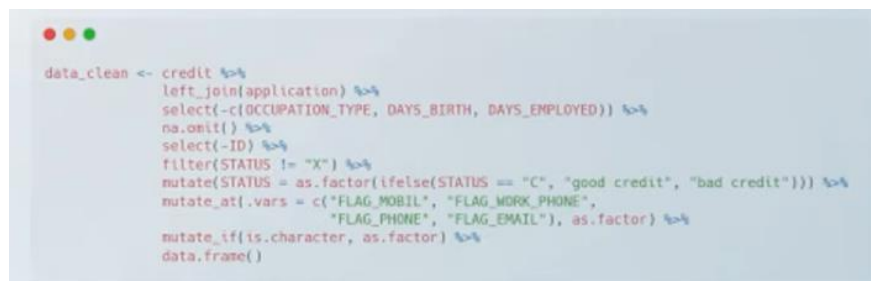
**Gambar 2. Kode Data Profil Nasabah**



```
#>      ID      CODE_GENDER  FLAG_OWN_CAR  FLAG_OWN_REALTY
#>      0              0              0              0
#>  CNT_CHILDREN  AMT_INCOME_TOTAL  NAME_INCOME_TYPE  NAME_EDUCATION_TYPE
#>      0              0              0              0
#>  NAME_FAMILY_STATUS  NAME_HOUSING_TYPE      DAYS_BIRTH      DAYS_EMPLOYED
#>      0              0              0              0
#>      FLAG_MOBIL  FLAG_WORK_PHONE      FLAG_PHONE      FLAG_EMAIL
#>      0              0              0              0
#>  OCCUPATION_TYPE  CNT_FAM_MEMBERS
#>      134203              0
```

Sumber: Algoritma Show

**Gambar 3. Missing Value**



```
data_clean <- credit %>%
  left_join(application) %>%
  select(-c(OCCUPATION_TYPE, DAYS_BIRTH, DAYS_EMPLOYED)) %>%
  na.omit() %>%
  select(-ID) %>%
  filter(STATUS != "X") %>%
  mutate(STATUS = as.factor(ifelse(STATUS == "C", "good credit", "bad credit"))) %>%
  mutate_at(vars = c("FLAG_MOBIL", "FLAG_WORK_PHONE",
                    "FLAG_PHONE", "FLAG_EMAIL"), as.factor) %>%
  mutate_if(is.character, as.factor) %>%
  data.frame()
```

Sumber: Algoritma Show

Cek *missing value* digunakan untuk mengetahui apakah data sudah lengkap atau ada data yang hilang. Ada dua cara dalam mengatasi *missing value*, yaitu (1) *take out variabel* atau membuang variabel-variabel yang memiliki jumlah *missing value* yang sangat besar (lebih dari 50%) dari total observasi dan (2) *complete case* adalah membuang baris-baris yang memiliki *missing value* karena jumlah observasi tidak terlalu banyak.

## 2. Menyesuaikan tipe data.

Untuk data-data yang kategorik yang sebelumnya memiliki tipe data karakter akan diubah menjadi tipe faktor.

**Gambar 4. Menyesuaikan Tipe Data**

```
data_clean <- credit %>%  
  left_join(application) %>%  
  select(-c(OCCUPATION_TYPE, DAYS_BIRTH, DAYS_EMPLOYED)) %>%  
  na.omit() %>%  
  select(-ID) %>%  
  filter(STATUS != "X") %>%  
  mutate(STATUS = as.factor(ifelse(STATUS == "C", "good credit", "bad credit"))) %>%  
  mutate_at(vars = c("FLAG_MOBIL", "FLAG_WORK_PHONE",  
                    "FLAG_PHONE", "FLAG_EMAIL"), as.factor) %>%  
  data.frame()
```

Sumber: Algoritma Show

## 3. *Exploratory data.*

**Gambar 5. Exploratory Data**

```
data_clean %>% inspect_cat() %>% show_plot()  
data_clean %>% inspect_num() %>% show_plot()
```

Sumber: Algoritma Show

Pada tahap ini akan dilakukan visualisasi data kategorik maupun data numerik.

**Gambar 6. Visualisasi Data**

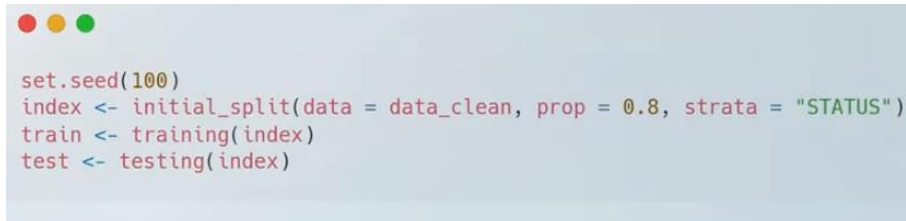


Sumber: Algoritma Show

Dari visualisasi data dapat dilihat bahwa status dari nasabah tersebut seimbang.

#### 4. *Cross-validation.*

**Gambar 7. Cross-Validation**

A screenshot of an R console window showing four lines of code for cross-validation. The code sets a seed, splits the data by status (80% train, 20% test), and then trains and tests a model on these splits.

```
set.seed(100)
index <- initial_split(data = data_clean, prop = 0.8, strata = "STATUS")
train <- training(index)
test <- testing(index)
```

Sumber: Algoritma Show

Pada tahap cross-validation dibagi menjadi dua bagian yang meliputi data *train* dan data *test*. Dari 80% data akan digunakan data *train* untuk modelling dan 20% data akan dijadikan data *test* sebagai evaluasi.

Setelah cleaning data, tahapan ketiga adalah modelling. Saat melakukan tahapan modelling dapat membandingkan beberapa model yang bisa digunakan seperti model random forest dan XGBoost. Random forest merupakan algoritma ensemble learning yang dibangun dari pohon keputusan menggunakan data bootstrap dan secara acak memilih subset variabel di setiap pohon keputusan. Kemudian, XGBoost merupakan algoritma yang powerful.

**Gambar 8. Model Random Forest**

A screenshot of an R console window showing code to create a Random Forest model using the caret package. It sets a seed, defines a training control with repeated cross-validation, and then trains the model.

```
set.seed(100)

ctrl <- trainControl(method = "repeatedcv",
                     number = 3,
                     repeats = 2,
                     allowParallel=FALSE)

model_forest <- caret::train(STATUS ~.,
                             data = train,
                             method = "rf",
                             trControl = ctrl)
```

Sumber: Algoritma Show

**Gambar 8. Model XGBoost**

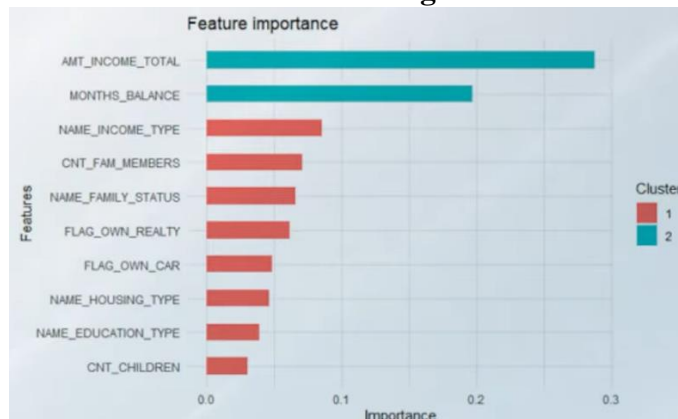
```
params <- list(booster = "gbtree",
               objective = "binary:logistic",
               eta=0.7,
               gamma=10,
               max_depth=10,
               min_child_weight=3,
               subsample=1,
               colsample_bytree=0.5)

xgbcv <- xgb.cv( params = params,
                 data = dtrain,
                 nrounds = 1000,
                 showsd = T,
                 nfold = 10,
                 stratified = T,
                 print_every_n = 50,
                 early_stopping_rounds = 20,
                 maximize = F)
```

Sumber: Algoritma Show

Setelah diimplementasikan, tahapan selanjutnya adalah membandingkan hasil dari kedua model tersebut. Model yang mana yang memiliki performance yang paling tinggi. Selanjutnya adalah tahap evaluasi model. Tahap evaluasi model adalah membandingkan matriks evaluasi dari model XGBoost dan Random forest. Dari kedua model tersebut, model XGBoost memiliki nilai yang lebih besar dibandingkan model random forest. Dari hasil model XGBoost dapat memperoleh informasi mengenai mana saja variabel-variabel yang paling berpengaruh dan paling penting di model tersebut.

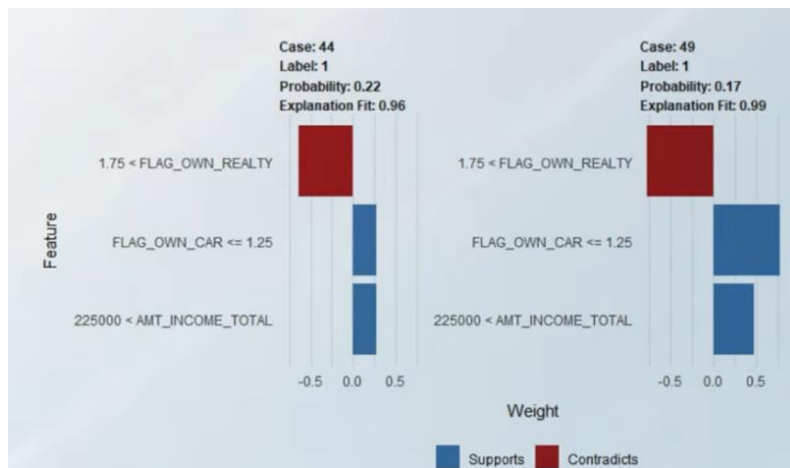
**Gambar 9. Variabel-Variabel Penting Dalam Credit Scoring**



Sumber: Algoritma Show

Pada gambar di atas terdapat 10 variabel yang paling penting apakah nasabah tersebut layak atau tidak layak. Variabel *annual income* atau total income nasabah menjadi variabel yang paling tinggi. Ini artinya variabel pendapatan menjadi paling penting untuk diprediksi, apakah nasabah layak atau tidak untuk diberikan pinjaman. Pada posisi kedua terdapat variabel *month balance*.

**Gambar 10. Probabilitas Kelayakan Nasabah**



Sumber: Algoritma Show

Melalui model ini diharapkan pihak bank dapat menggunakan model tersebut untuk mengetahui berapa probabilitas nasabah tersebut layak atau tidak diberikan pinjaman.

## Lampiran

Link Gthub: [Vinahasan.github.io](https://github.com/Vinahasan)

## Daftar Pustaka

Ajeng. Credit Scoring. Youtube, diunggah oleh Algoritma Show 12 Juli 2021, <https://youtu.be/L7564DMRcdY>.