

# Final Project

## Loan Application Prediction Model (2007 – 2014)

Presented by

Ghina Hanifah Alamsyah

Link ZIP: [https://drive.google.com/file/d/18tW-ApK\\_Ik3i7UOpqwGaCh6sKMR\\_cfJC/view?usp=drive\\_link](https://drive.google.com/file/d/18tW-ApK_Ik3i7UOpqwGaCh6sKMR_cfJC/view?usp=drive_link)

# About the Company

ID/X Partners didirikan pada tahun 2006 yang **merupakan perusahaan konsultan** yang didirikan oleh para profesional berpengalaman di bidang perbankan dan manajemen. Memiliki berbagai keahlian pada **bidang pengelolaan siklus kredit, pengembangan scoring, dan manajemen kinerja** yang telah digunakan oleh berbagai perusahaan, termasuk di sektor keuangan, telekomunikasi, manufaktur, dan ritel.

Selain itu, ID/X Partners **mengandalkan pendekatan berbasis data analytics dan decisioning (DAD)** yang terintegrasi dengan manajemen risiko dan strategi pemasaran untuk meningkatkan profitabilitas dan efisiensi bisnis klien.

# Objectives

Adapun tujuan dari *project* ini yaitu membangun model yang dapat memprediksi credit risk menggunakan dataset yang telah disediakan oleh company, yang terdiri dari kategori data pinjaman yang **diterima** dan yang **ditolak**.

# Data Understanding

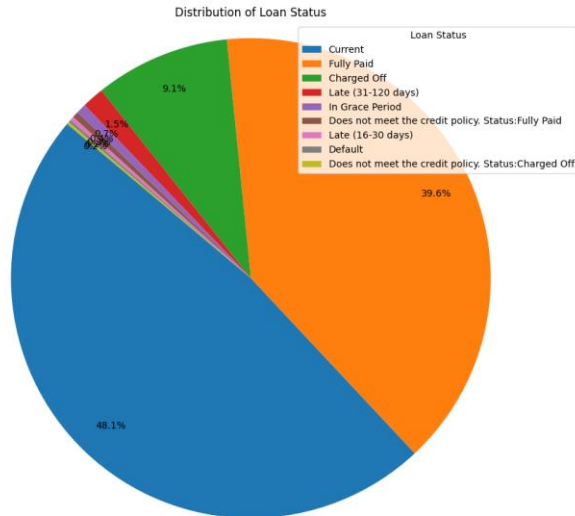
```
# Define the whole dataset as 'loan'
loan = pd.read_csv("loan_data_2007_2014.csv")

... /tmp/ipython-input-2469554926.py:2: DtypeWarning: Co
    loan = pd.read_csv("loan_data_2007_2014.csv")

# Find out the shape of the dataset (rows, columns)
loan.shape

(466285, 75)
```

- Project ini menggunakan dataset dari file “loan\_data\_2007\_2014.csv”;
- Dataset terdiri dari 466.285 row data dan 75 kolom fitur;
- Dataset terdiri dari 53 data numerik dan 22 data kategorik;
- Distribusi dari status pinjaman para nasabah selama 2007-2014 didominasi oleh status “*current*” (sedang dalam masa pembayaran) dan “*fully paid*” (sudah lunas).



# Data Preprocessing

```
# Drop the irrelevant columns with missing values

columns_to_drop = ['id', 'member_id', 'sub_grade', 'emp_title', 'url', 'desc', 'title', 'zip_code', 'next_pymnt_d',
                   'recoveries', 'collection_recovery_fee', 'total_rec_prncp', 'total_rec_late_fee', 'desc', 'mths_since_last_record',
                   'mths_since_last_major_derog', 'annual_inc_joint', 'dti_joint', 'verification_status_joint', 'open_acc_6m', 'open_il',
                   'open_il_12m', 'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il', 'il_util', 'open_rv_12m', 'open_rv_24m',
                   'max_bal_bc', 'all_util', 'inq_fi', 'total_cu_tl', 'inq_last_12m', 'policy_code',]

loan.drop(columns=columns_to_drop, inplace=True, axis=1)

loan.dropna(inplace=True)
```

```
loan.duplicated().sum()
```

```
np.int64(0)
```

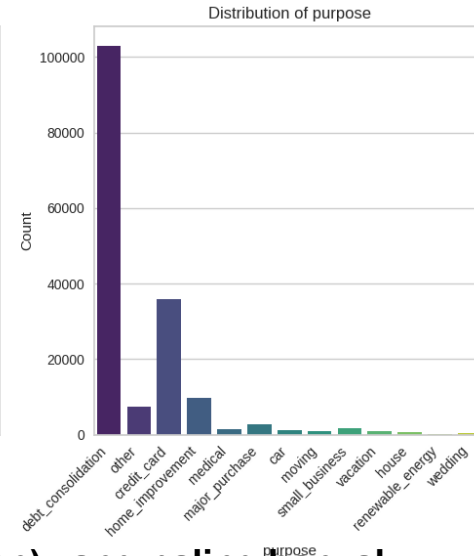
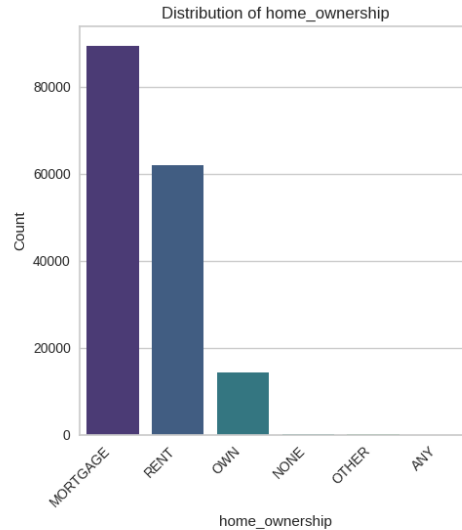
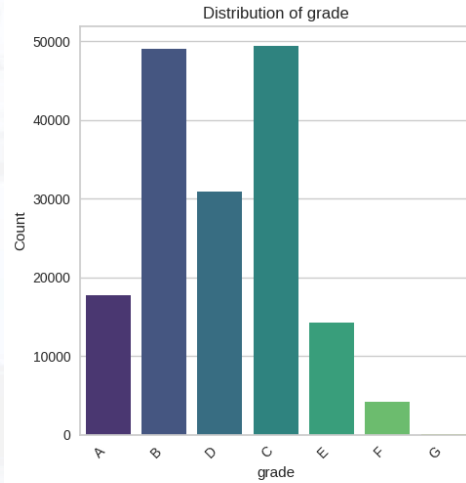
- Kolom-kolom yang memiliki *missing values* > 50% akan dibuang;
- Baris data (*row*) yang memiliki nilai NaN juga akan dibuang untuk pembersihan data;
- Kolom-kolom (*features*) yang mengandung identitas peminjam dan data tentang masa depan (yang belum diketahui) juga perlu dibuang karena sifatnya belum terjadi;
- Tidak ada data yang bersifat duplikat.

# Data Preprocessing



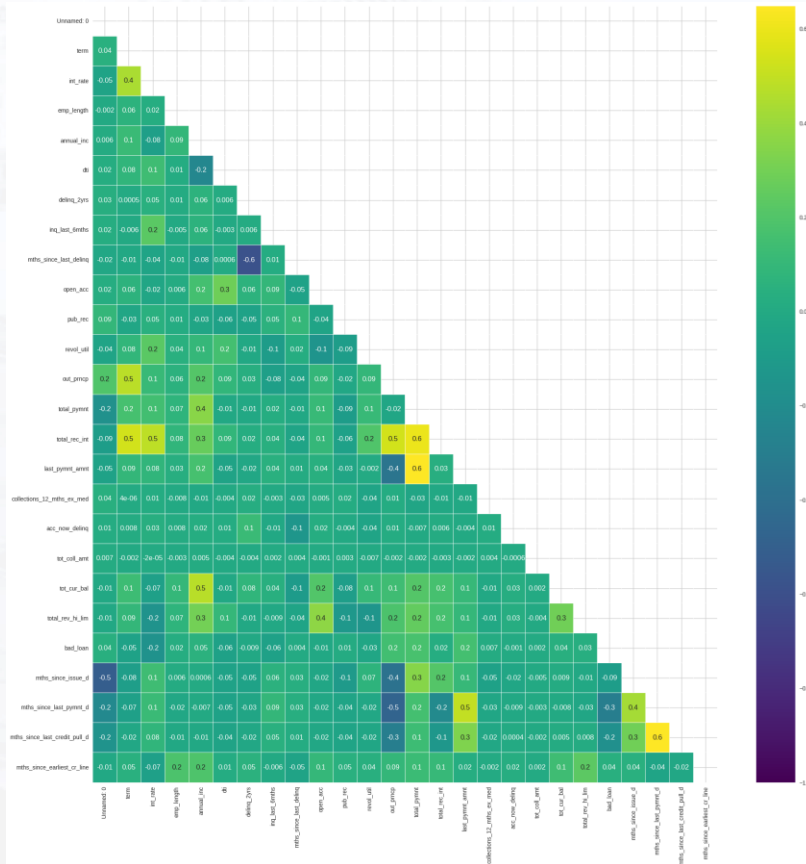
- Menghapus data-data pada *features* tertentu (`int_rate`, `annual_inc`, `dti`, dan `open_acc`) yang memiliki *outliers*
- Alasan mengapa keempat *features* ini yang dipilih adalah:
  - **int\_rate:** Berpengaruh terhadap besar pinjaman dan dapat menjadi indikator resiko atau kondisi pasar
  - **annual\_inc:** Kemampuan peminjam untuk membayar pinjaman
  - **dti:** Menilai beban utang yang dimiliki peminjam yang berhubungan dengan pendapatan mereka
  - **open\_acc:** Menunjukkan *credit behavior* peminjam dan akses kredit.

# Exploratory Data Analysis (EDA)



- Berdasarkan plot tersebut, **grade (tingkat resiko pinjaman) yang paling banyak didapat oleh peminjam adalah grade B dan C**, di mana semakin ke kiri, resiko dan tingkat bunga semakin rendah, dan sebaliknya;
- **Tipe kepemilikan rumah terbanyak berupa mortgage** (asset tidak bergerak seperti rumah, apartemen, dll.);
- **Tujuan pinjaman yang paling banyak diajukan oleh peminjam adalah konsolidasi utang (debt consolidation).**

# Exploratory Data Analysis (EDA)



- tot\_cur\_bal (saldo total sekarang dari semua akun) berkorelasi dengan annual\_inc (total pendapatan peminjam per tahun) sebesar 0.5;
- int\_rate (tingkat bunga) berkorelasi dengan term (jumlah pembayaran pinjaman) sebesar 0.5;
- total\_rec\_int (bunga yang diterima hingga saat ini) berkorelasi dengan int\_rate dan juga term sebesar 0.5, serta dengan total\_pymnt (pembayaran yang diterima sampai saat ini untuk jumlah total yang didanai) sebesar 0.6.

# Features Engineering



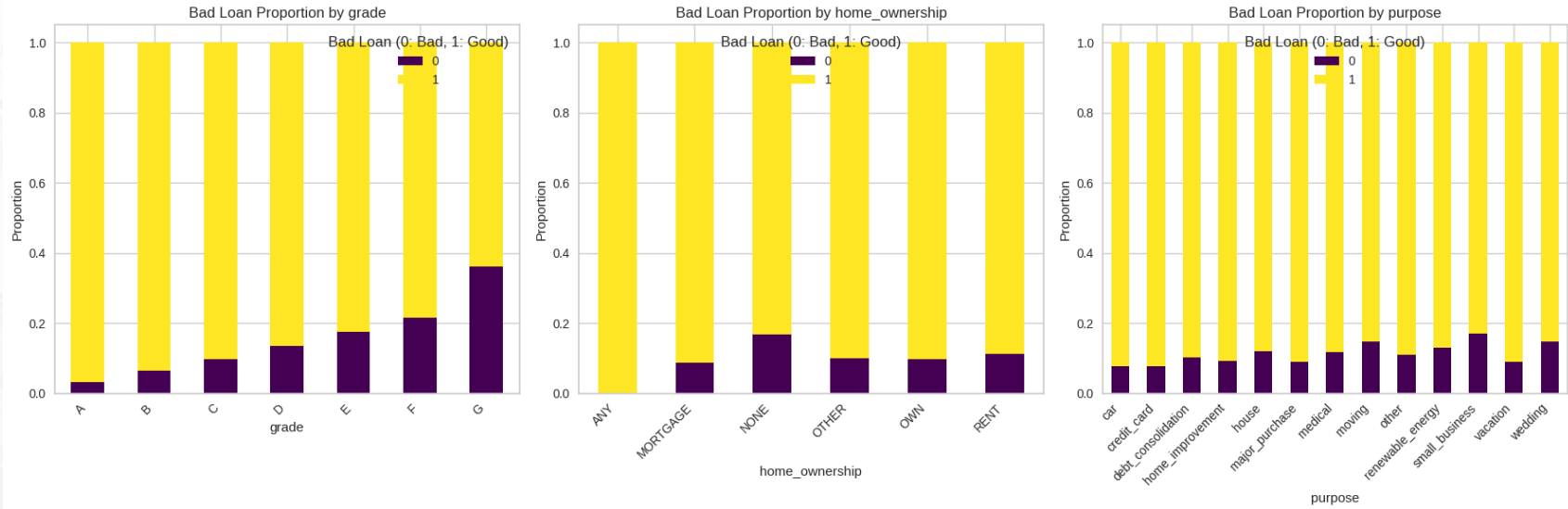
```
#dealing with imbalanced data
from imblearn.over_sampling import RandomOverSampler
os = RandomOverSampler()
X_train_o, y_train_o = os.fit_resample(X_train, y_train)
y_train_series = pd.Series(y_train_o)
#check value counts after oversampling
y_train_series.value_counts()
```

```
...          count
bad_loan
1          119606
0          119606

dtype: int64
```

- Untuk menyeimbangkan distribusi data berkategori 0 (*bad loan*) dan 1(*good loan*), dilakukan teknik *oversampling*;
- Data yang bernilai '*Charged Off*', '*Default*', '*Late (31-120 days)*', '*Does not meet the credit policy. Status: Charged Off*', dikategorikan sebagai 0 (*bad loan*).

# Features Engineering



- Persebaran *bad loan* dan *good loan* pada fitur-fitur *grade*, *home ownership*, dan *loan purpose*.

# Building Models and Evaluation

## Logistic Regression

```
#predicting
from sklearn.metrics import classification_report
y_preds = model.predict(X_test_encoded)
#classification report
print(classification_report(y_test, y_preds))
```

```
...      precision    recall  f1-score   support

      0       0.58       0.86       0.69       3300
      1       0.98       0.93       0.96      29817

 accuracy          0.92       33117
 macro avg       0.78       0.89       0.83       33117
 weighted avg    0.94       0.92       0.93       33117
```

## Random Forest

```
from sklearn.metrics import classification_report

# Generate classification report for Random Forest model
print("Classification Report for Random Forest Model:")
print(classification_report(y_test, y_preds_rf))
```

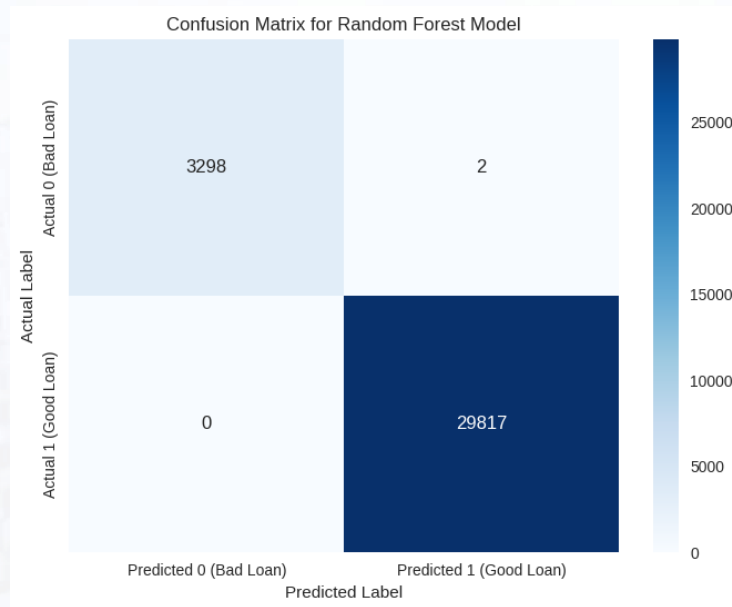
```
... Classification Report for Random Forest Model:
      precision    recall  f1-score   support

      0       1.00       1.00       1.00       3300
      1       1.00       1.00       1.00      29817

 accuracy          1.00       33117
 macro avg       1.00       1.00       1.00       33117
 weighted avg    1.00       1.00       1.00       33117
```

- Menggunakan dua model, yaitu Logistic Regression dan Random Forest. Namun dapat dilihat bahwa **Random Forest lebih unggul dalam akurasi dan presisi.**

# Building Models and Evaluation



Kedua model cenderung lebih mampu memprediksi *good loan* sebagai *good loan*, yang tergolong ke *True Positive* (TP). Hanya sedikit *False Positive* (FP) dan *False Negative* (FN) yang diprediksi.

# Model Testing

```
pd.Series(y_preds_rf).value_counts() # Find out how many good and bad loans from the customers
```

	count
1	29819
0	3298

dtype: int64

Dengan menggunakan model Random Forest, dapat ditentukan bahwa **peminjam yang mendapatkan *bad loan* ada sebanyak 3298 orang. Mayoritas jenis pinjaman tergolong ke *good loan*.**

## Kesimpulan:

- Mayoritas pinjaman para peminjam tergolong pada *good loan*;
- Model *Random Forest* lebih cocok digunakan untuk kasus ini;

## Rekomendasi untuk Perusahaan:

- **Kembangkan kelengkapan data:** Kurangi nilai-nilai yang hilang, terutama untuk fitur-fitur yang krusial untuk menentukan prediksi. Hal ini dapat mencakup validasi input yang lebih ketat atau pedoman yang lebih jelas bagi petugas pinjaman;
- **Data Audit:** Secara teratur lakukan audit data untuk masalah kualitasnya, termasuk outliers dan ketidakkonsistennya, untuk meyakinkan seberapa *reliable* insight yang diperoleh dari analisis datanya.

# Thank You



**Rakamin**  
Academy



id/x partners