Rakamin Academy

PAYMENT DEFAULT PREDICTION AT HOMECREDIT

Dokumen Laporan Final Project

Kelompok 1 - Octagram

OCTAGRAM'S TEAM

Team Leader

Aldi Vibriani

Business Intelligence

Andiny Lusy Septiariany

Nur Alifa Zahra

Data Analyst

David Yudha Prasetya

Ramadhian Ekaputra

Data Scientist

Alfriando C Vean

Reza Yulhansyah

Latar Belakang Masalah

Home Credit adalah perusahaan layanan keuangan yang menyediakan solusi pembiayaan konsumen. Didirikan pada tahun 1997, Home Credit awalnya beroperasi di Republik Ceko dan sejak itu telah berkembang ke berbagai negara di seluruh dunia, termasuk Indonesia.

Perusahaan ini dikenal dengan produk pinjaman yang mudah diakses, seperti kredit tanpa agunan, cicilan barang elektronik, dan pembiayaan untuk berbagai kebutuhan konsumen.

Latar Belakang Masalah

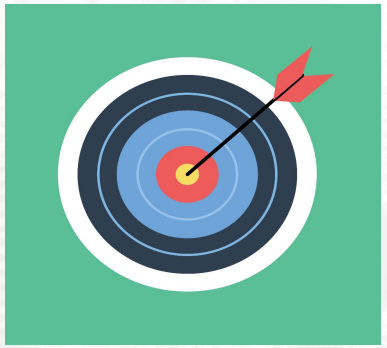
Menurut survey dari *Survey of Consumer Expectations (SCE) Credit Access Survey*, terjadi peningkatan penolakan pengajuan pinjaman kredit dari 0.9% menjadi 21% di Tahun 2024 (newyorkfed.org, 2024). Salah satu yang menjadi penyebab pengajuan ditolak adalah kurangnya credit history customer dan ketatnya spesifikasi yang harus dipenuhi agar pengajuan pinjaman dapat diterima (rocketmortgage, 2023., Backendsnews, 2022).

Latar Belakang Masalah

Maka dari itu, Home Credit memiliki misi untuk memberikan pinjaman yang responsible, reliable, dan affordable untuk setiap orang, termasuk customer yang memiliki *lack of credit history* (homecredit.co.in, 2024).

Berdasarkan beberapa penelitian yang telah dilakukan, untuk dapat memprediksi seseorang itu default (gagal bayar) atau tidak adalah dengan menerapkan machine learning (Bazzana F, 2023., datarobots, n.d., Liu Y, 2022., Fico, 2022)

Latar Belakang Masalah



Goals

Memprediksi pengajuan customer apakah mereka **default** atau **tidak** (gagal bayar atau tidak)

Objective

Mengimplementasikan **machine learning** untuk mencari model terbaik dalam **memprediksi** customer yang **default**

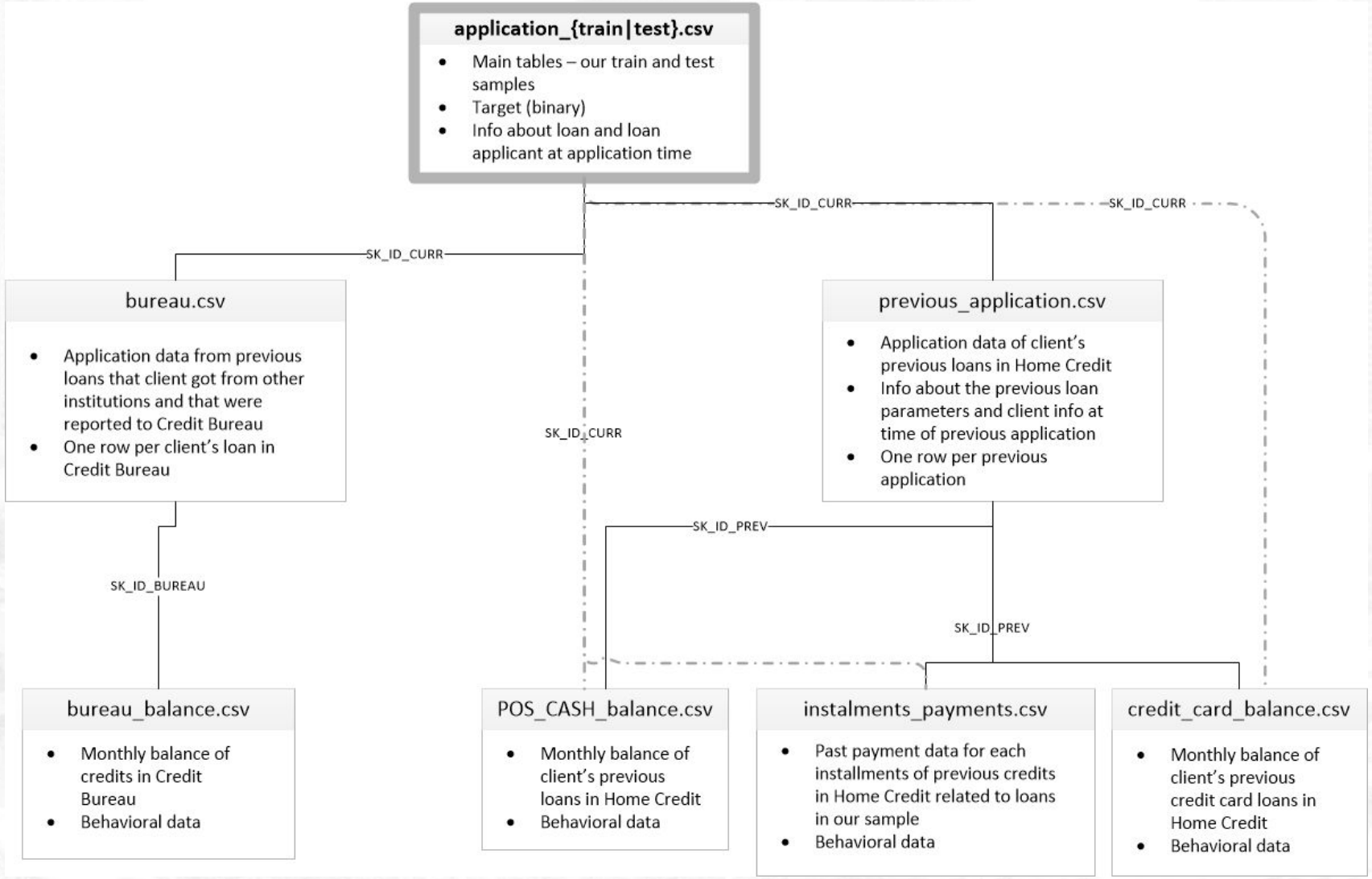
Business Metrics

Business metric yang digunakan adalah **Credit Default Rate**, yatu persentase seseorang default atau tidak, diukur dengan membandingkan sebelum dan sesudah model diuji.

Exploratory Data Analysis

Rakamin Academy
School for Career Acceleration

Home Credit Datasets

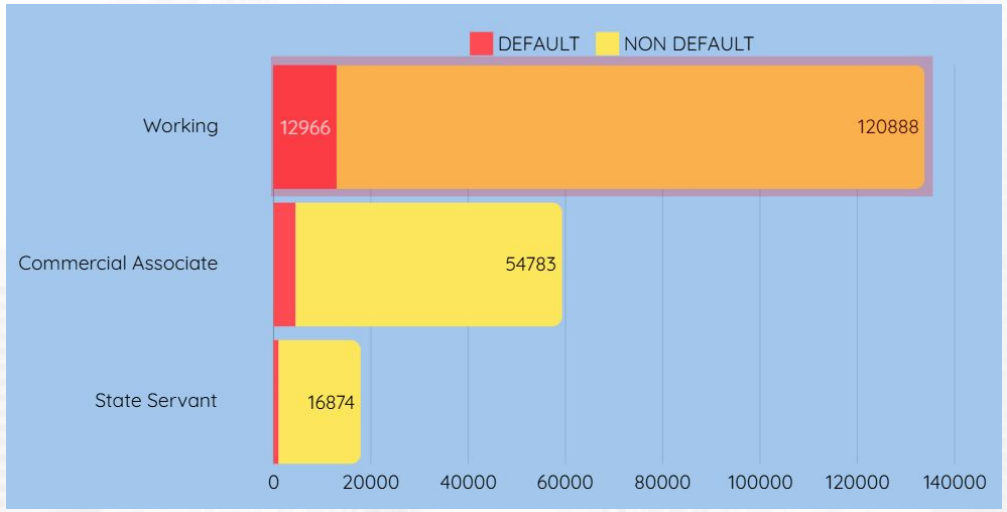


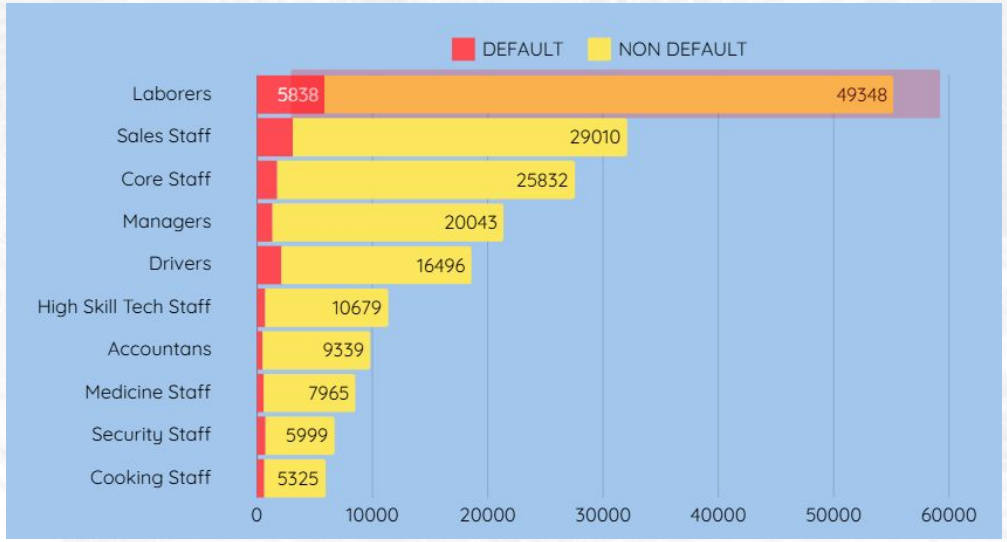
Exploratory Data Analysis

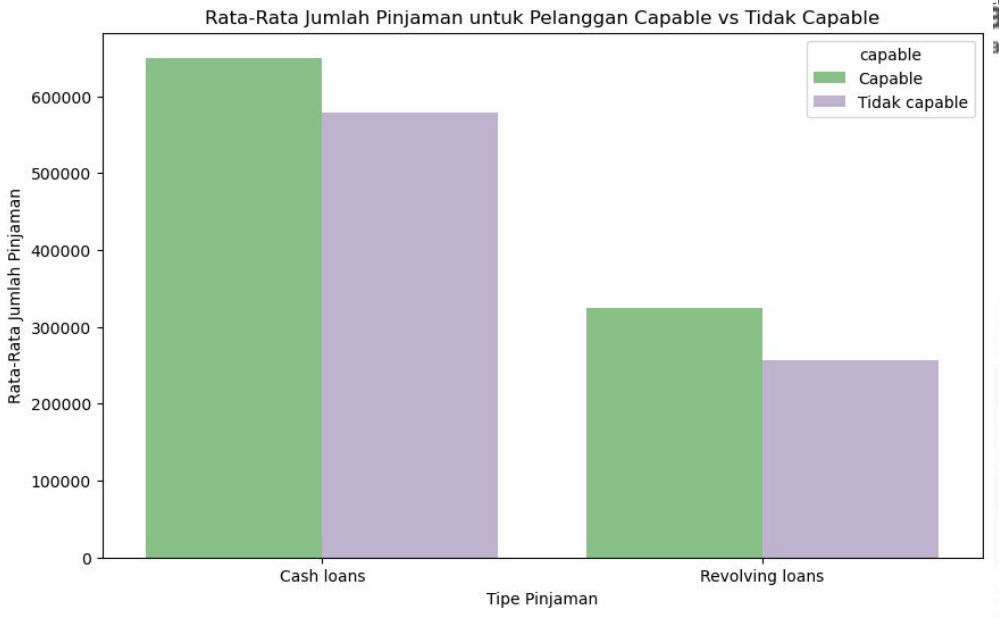
Terdapat 8 Datasets yang kami gunakan untuk pemodelan machine learning. Langkah - langkah kami lakukan antara lain:

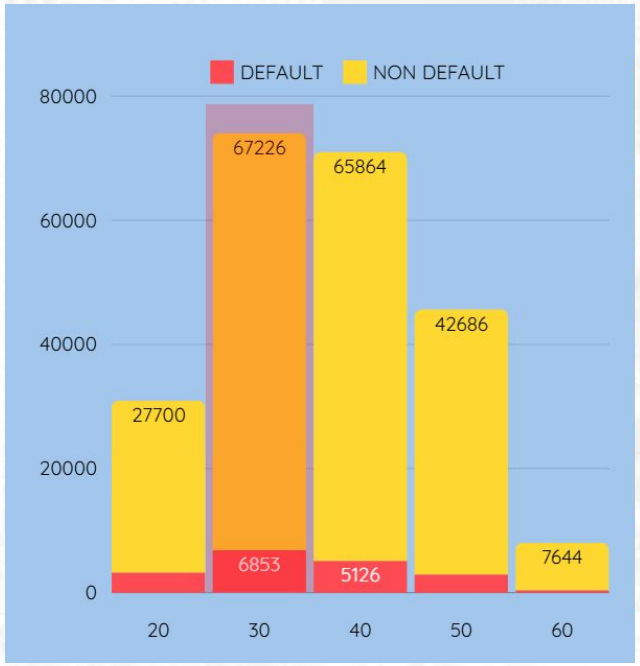
- Melakukan ***Exploratory Data Analysis*** **(EDA)** untuk memahami masing - masing dataset dan *features*-nya.
- Lalu kami melakukan **df.describe(include='all')** untuk memahami keseluruhan features. Apakah terdapat outliers dari mean dan std. deviasi, melihat nilai minimum dan maksimumnya (untuk numerical data) dan nilai unique dan modus (untuk categorical data).
- Kami juga melakukan visualisasi menggunakan **histogram** untuk melihat skewness dan **boxplot** untuk melihat outliers. Hasil menunjukan bahwa cukup banyak features yang skew baik itu positif maupun negatif dan juga outliers (positif dan negatif outliers).
- Kami juga melakukan cek korelasi menggunakan **heatmap** dengan metode **spearman** untuk melihat korelasi antar feature dan korelasi feature dengan target karena **spearman robust terhadap outliers**. Hasil menunjukan **tidak feature yang berkorelasi dengan target**. Hal ini terjadi karena feature merupakan categorical (dalam boolean) sehingga untuk pemodelan akan digunakan semua feature dan diperkecil menggunakan selectKBest karena metode tersebut dapat mengetahui feature apa saja yang memiliki kekuatan (atau seberapa penting) dalam mempengaruhi target.

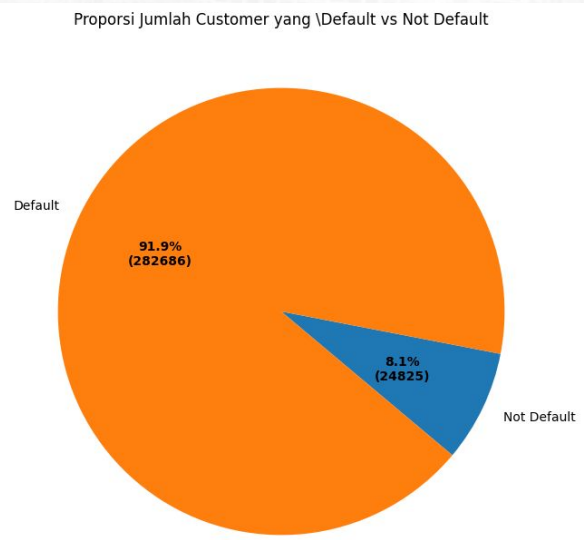
Business Insight











Business Insight

Mengingat banyaknya features dan dataset yang digunakan, kami akan menjelaskan beberapa feature yang kami anggap menarik sebagai insight, antara lain:

- Label "TARGET" menjelaskan bahwa sebanyak 91.9% customer itu tidak default dan 8.1% customer termasuk default.
- Berdasarkan tipe pinjaman yang paling banyak dicari adalah Cash Loans dibandingkan Revolving Loans dan secara rata - rata, orang yang meminjam dengan tipe Cash Loans tinggi untuk default.
- Secara usia, rata - rata peminjam paling tinggi berada di kisaran usia 30 - 40 tahun dan orang yang sering default berada di kisaran usia 30 tahun. Hal ini perlu menjadi pertimbangan Home Credit karena rentang usia 30 tahun menjadi peminjam terbanyak sekaligus paling sering default juga.
- Berdasarkan pendapatan, sumber penghasilan utama customer adalah dengan bekerja sebagai buruh dan hasil menunjukan juga bahwa buruh sering sekali default sehingga Home Credit perlu berhati - hati untuk customer ini.

note: Terdapat beberapa insight yang tidak dapat ditulis karena keterbatasan jumlah maksimum slide pada business insight

Pre-processing (Data Cleansing)

Dataset	Total Rows	Total Columns	Jumlah Columns Dengan Null Value	Jumlah Columns Dengan Duplikat Data
Application train	307.511	122	70	0
Application test	48.744	121	70	0
Bureau	1.716.428	17	9	0
Bureau Balance	27.299.925	3	0	0
Previous Application	1.670.214	37	26	0
Credit Card Balance	3.840.312	20	9	0
Posh Cash Balance	10.001.358	8	3	0
Installment	13.605.401	8	2	0

note: Preprocessing final ini kami lakukan setelah beberapa kali percobaan karena sebelumnya model sulit menemukan angka terbaik.

Berikut adalah detail dari masing - masing datasets dan proses preprocessing yang kami lakukan:

- Menggunakan **df.info()**, kami dapat melihat jumlah data, columns, null value, dan tipe data setiap columns.
- Beberapa feature yang memiliki null value berada di **atas 40%** karena kami anggap tidak memberikan insight dan dapat mengurangi jumlah features yang banyak, lalu untuk yang kami anggap penting kami isi dengan 0, dan di **bawah 40%** kami isi dengan median dan modus karena kami tidak ingin kehilangan data penting, lebih robust, dan tidak terlalu mengubah keseluruhan data.
- T**idak ada data duplikat** yang kami temukan.
- Terdapat beberapa **outliers** tapi kami anggap masih diterima dan agar model dapat mempelajari outliers pada data baru yang lebih general nanti.

Pre-processing (Feature Encoding)

Rakamin Academy
School for Career Acceleration

Dari delapan dataset, terdapat beberapa feature categorical yang perlu diubah menjadi numeric agar bisa diolah di machine learning. Namun, kami melakukan label encoding saja untuk feature categorical yang memiliki ciri - ciri ordinal atau hanya dua value saja. Untuk feature yang tidak memenuhi kriteria tersebut kami hapus terlebih dahulu karena untuk mengurangi size data saat penggabungan agar proses pemodelan machine learning bisa lebih cepat.

1. **Application Train& Test**

```
# Feature Encoding
def feature_encode(data):
    data['FLAG_OWN_CAR'] = data['FLAG_OWN_CAR'].map({'N': 0, 'Y': 1})
    data['FLAG_OWN_REALTY'] = data['FLAG_OWN_REALTY'].map({'N': 0, 'Y': 1})
    data['WEEKDAY_APPR_PROCESS_START'] = data['WEEKDAY_APPR_PROCESS_START'].map({'MONDAY': 1, 'TUESDAY': 2, 'WEDNESDAY': 3, 'THURSDAY': 4, 'FRIDAY': 5, 'SATURDAY':
    data['NAME_EDUCATION_TYPE'] = data['NAME_EDUCATION_TYPE'].map({'Lower secondary': 0, #Sekolah Menengah Pertama
                                                                   'Secondary / secondary special': 1, #Sekolah Menengah Atas
                                                                   'Incomplete higher': 2, # Pendidikan Tinggi yang belum selesai
                                                                   'Higher education': 3, # Pendidikan Tinggi
                                                                   'Academic degree': 4}) # Gelar Akademik
```

[17] Python

Pada application_train & test terdapat feature boolean dan ordinal seperti WEEKDAY_APP_PROCESS_START dan NAME_EDUCATION_TYPE sehingga kami ubah secara manual menggunakan .map() untuk secara spesifik mengurutkan nama - nama value tersebut.

2. **Bureau & Bureau Balance**

```
# Feature Encoding
def bureau_merge_encoding(data):
    bureau_merge['CREDIT_CURRENCY'] = bureau_merge['CREDIT_CURRENCY'].map({'currency 1': 1, 'currency 2': 2, 'currency 3': 3, 'currency 4': 4})
```

Setelah menggabungkan Bureau dan Bureau Balance berdasarkan SK_ID_BURR, kami menemukan satu feature categorical yang kami anggap bisa diubah menjadi angka yang berurutan.

Pre-processing (Feature Encoding)

Rakamin Academy
School for Career Acceleration

3. Previous Application

```
# Feature Encoding
def prev_encoding(data):
    data['WEEKDAY_APPR_PROCESS_START'] = data['WEEKDAY_APPR_PROCESS_START'].map({'MONDAY': 1, 'TUESDAY': 2, 'WEDNESDAY': 3, 'THURSDAY': 4, 'FRIDAY': 5,
    data['FLAG_LAST_APPL_PER_CONTRACT'] = data['FLAG_LAST_APPL_PER_CONTRACT'].map({'N': 0, 'Y': 1})
    data['NAME_CASH_LOAN_PURPOSE'] = data['NAME_CASH_LOAN_PURPOSE'].replace({'unknown': 'XAP'}) # Menggabungkan unknown dengan XAP
    data['NAME_YIELD_GROUP'] = data['NAME_YIELD_GROUP'].map({'unknown': 0, 'low_action': 1, 'low_normal': 2, 'middle': 3, 'high': 4})
    prev['NAME_CASH_LOAN_PURPOSE'] = prev['NAME_CASH_LOAN_PURPOSE'].replace({
                                                    'Education': 'Other', 'Journey': 'Other', 'Purchase of electronic equipment': 'Other
                                                    'Wedding / gift / holiday': 'Other', 'Buying a home': 'Other', 'Car repairs': 'Other
                                                    'Buying a holiday home / land': 'Other', 'Business development': 'Other', 'Gasificat
                                                    'Buying a garage': 'Other', 'Hobby': 'Other', 'Money for a third person': 'Other', '
    })
    data['NAME_GOODS_CATEGORY'] = data['NAME_GOODS_CATEGORY'].map({
                                                    # Electronics group
                                                    'Mobile': 'Electronics',
                                                    'Consumer Electronics': 'Electronics',
                                                    'Computers': 'Electronics',
                                                    'Audio/Video': 'Electronics',
                                                    'Photo / Cinema Equipment': 'Electronics',
                                                    'Office Appliances': 'Electronics',

                                                    # Home and Living group
                                                    'Furniture': 'Home and Living',
                                                    'Construction Materials': 'Home and Living',
                                                    'Homewares': 'Home and Living',
                                                    'Gardening': 'Home and Living',
                                                    'House Construction': 'Home and Living',
```

Previous Application memiliki feature categorical yang bisa dilakukan label encoding. Terdapat feature ordinal dan boolean yang kami ubah menjadi angka yang berurutan. Kemudian, ada feature yang yang kami categorical untuk di One Hot Encoding agar saat OHE tidak terlalu memunculkan banyak feature. Namun, pada proses modelling, kami tidak menggunakan feature OHE terlebih dahulu karena data cukup berat saat di jalankan.

Pre-processing (Feature Extraction)

Kami melakukan feature extraction untuk beberapa dataset sebagai berikut:

1. **Application Train & Test**

```python
def application_FE(data):

    # Income Ratio
    data['DEBT_TO_INCOME'] = data['AMT_CREDIT'] / data['AMT_INCOME_TOTAL'] # Untuk mengukur seberapa mampu nasabah akan membayar dari total pendapatannya
    data["ANNUITY_TO_INCOME"] = data["AMT_ANNUITY"] / data["AMT_INCOME_TOTAL"] # Untuk mengukur rasio angsuran bulanan terhadap pendapatan nasabah

    # Usia dan Tenure
    data['AGE'] = round(-data['DAYS_BIRTH'] / 365).astype('Int64')
    data['TENURE'] = round(-data['DAYS_EMPLOYED'] / 365).astype('Int64') # Mengukur berapa tahun lamanya nasabah bekerja

    # Keluarga
    data['FAM_ADULT_MEMBERS'] = data['CNT_FAM_MEMBERS'] - data['CNT_CHILDREN'] # Menghitung berapa orang dewasa dalam satu keluarga
    data['CHILDREN_PER_FAM_MEMBERS'] = data['CNT_CHILDREN'] / data['CNT_FAM_MEMBERS'] # Menghitung rasio jumlah potensi tanggungan yang dimiliki

    # Kelengkapan Document
    data['NUM_DOCUMENTS'] = data.filter(like='FLAG_DOCUMENT').sum(axis=1)

    # External Source Mean
    data["EXT_SOURCE_MEAN"] = data[["EXT_SOURCE_1", "EXT_SOURCE_2", "EXT_SOURCE_3"]].mean(axis = 1)
```

Di sini kami membuat beberapa feature yang sekiranya bisa mengurangi curse of dimension dan beberapa feature yang memiliki korelasi yang sangat tinggi. Alhasil, feature tersebut kami buat sesuai gambar di atas. Selain application train dan test, kami juga melakukan feature extraction di dataset – dataset lain seperti di slide selanjutnya.

Pre-processing (Feature Extraction)

Kami melakukan feature extraction untuk beberapa dataset sebagai berikut:

2. Bureau & Bureau Balance

```
# Feature Engineering
def bureau_merge_FE(data):

    # Durasi lamanya pinjaman
    data['DAYS_DURATION_CREDIT'] = data['DAYS_ENDDATE_FACT'] - data['DAYS_CREDIT']
```

Karena bureau dan bureau balance tidak terlalu banyak feature, kami hanya menemukan ide untuk membuat feature days duration credit dimana feature ini memberi tahu kita berapa lamanya pinjaman customer di bank lain (dalam hitungan hari).

Pre-processing (Feature Extraction)

Kami melakukan feature extraction untuk beberapa dataset sebagai berikut:

3. Installment

```python
# Feature Engineering
def installment_FE(data):
    # Melihat keterlambatan pembayaran
    data['DAYS_LATE'] = data['DAYS_ENTRY_PAYMENT'] - data['DAYS_INSTALMENT'] # Nilai positif menunjukan keterlambatan pembayaran

    # Ratio pembayaran yang diharapkan vs aktual
    data['PAYMENT_RATIO'] = data['AMT_PAYMENT']/data['AMT_INSTALMENT'] # Nilai kurang dari satu menunjukan pembayaran yang kurang

    # Melihat Pembayaran yang Kurang
    data['PAYMENT_DIFF'] = data['AMT_INSTALMENT'] - data['AMT_PAYMENT'] # Jika hasilnya positif menunjukan bahwa ada pembayaran yang kurang

    # Pembagian waktu
    data['MONTH_INSTALMENT'] = round(data['DAYS_INSTALMENT'] / 12)
    data['MONTH_ENTRY_PAYMENT'] = round(data['DAYS_ENTRY_PAYMENT'] / 12)
    data['YEAR_INSTALLMENT'] = round(data['DAYS_INSTALMENT'] / 365)
    data['YEAR_ENTRY_PAYMENT'] = round(data['DAYS_ENTRY_PAYMENT'] / 365)

    # Flag
    data['FLAG_LATE_PAYMENT'] = (data['DAYS_LATE'] > 0).astype(int) # Jika days_late atau month_installment lebih kecil dibandingkan month_entry_payment menunjukan
    data['FLAG_UNDERPAYMENT'] = (data['PAYMENT_RATIO'] > 1).astype(int)
```

Dataset ketiga adalah installment dimana kami cukup banyak membuat feature extraction seperti gambar di atas untuk melihat keterlambatan pembayaran dalam hari, bulanan, dan tahunan, lalu melihat ratio pembayaran, apakah ada pembayaran yang belum lunas, sehingga akan ditandai (flag) bagi mereka yng telat atau pembayaran tidak lunas (under payment).

Pre-processing (Feature Extraction)

Kami melakukan feature extraction untuk beberapa dataset sebagai berikut:

3. Credit Card Balance

```python
# Feature Engineering
def credit_card_balance_FE(data):

    # Ratio limit kredit terpakai
    data['LIMIT_RATIO'] = data['AMT_BALANCE'] / data['AMT_CREDIT_LIMIT_ACTUAL']

    # Total Penarikan dari Semua Channel
    data['TOTAL_DRAWINGS'] = (data['AMT_DRAWINGS_ATM_CURRENT'] +
                              data['AMT_DRAWINGS_CURRENT'] +
                              data['AMT_DRAWINGS_OTHER_CURRENT'] +
                              data['AMT_DRAWINGS_POS_CURRENT'])

    # Total Frekuensi Penarikan dari Semua Channel
    data['TOTAL_DRAWINGS_COUNT'] = (data['CNT_DRAWINGS_ATM_CURRENT'] +
                                    data['CNT_DRAWINGS_CURRENT'] +
                                    data['CNT_DRAWINGS_OTHER_CURRENT'] +
                                    data['CNT_DRAWINGS_POS_CURRENT'])

    # Rata - Rata Jumlah Uang yang ditarik per Penarikan
    data['AVG_DRAWING'] = data['TOTAL_DRAWINGS'] / data['TOTAL_DRAWINGS_COUNT']

    # Percentase Penarikan di setiap Channel
    data['DRAWINGS_ATM_PERCENTAGE'] = data['AMT_DRAWINGS_ATM_CURRENT'] / data['TOTAL_DRAWINGS']
    data['DRAWINGS_POS_PERCENTAGE'] = data['AMT_DRAWINGS_POS_CURRENT'] / data['TOTAL_DRAWINGS']
    data['DRAWINGS_OTHER_PERCENTAGE'] = data['AMT_DRAWINGS_OTHER_CURRENT'] / data['TOTAL_DRAWINGS']

    # Flag Pembayaran yang Telat
    data['FLAG_LATE_PAYMENT'] = (data['NAME_CONTRACT_STATUS'] == 'Late').astype(int)

    # Handling infinite value
    data = data.replace([np.inf, -np.inf], np.nan)
```

Selanjutnya adalah feature extraction pada credit card balance dimana kami ingin melihat behavior customer dalam menggunakan kartu kredit seperti limit ratio mereka, berapa jumlah penarikan dan saldo yang ditarik oleh customer, serta ingin melihat siapa saja yang membayar telat agar bisa mendapat insight ciri - ciri customer yang sering telat membayar.

Pre-processing (Feature Extraction)

Kami melakukan feature extraction untuk beberapa dataset sebagai berikut:

4. **Pos Cash Balance**

```python
# Feature Engineering
def pos_cash_balance_FE(data):

    # Menghitung Ratio Cicilan yang Berjalan
    data['PAYMENT_PROGRESS_RATIO'] = data['CNT_INSTALMENT'] / data['CNT_INSTALMENT_FUTURE']

    # Menghitung Sisa Cicilan
    data['REMAINING_PAYMENT'] = data['CNT_INSTALMENT_FUTURE'] - data['CNT_INSTALMENT']
```

Tidak terlalu banyak feature yang bisa di explore pada Point of Sales (POS) Cash Balance sehingga kami membuat feature untuk melihat ratio cicilan yang dibayarkan pada Pos Cash Balance ini dan melihat berapa sisa cicilan yang harus dibayarkan untuk melihat perilaku customer apakah mampu untuk melunasi cicilan atau tidak.

Previous application lebih banyak feature categorical sehingga tidak ada feature extraction yang bisa kami lakukan. Kami lebih memfokuskan untuk mengecek unique values, tipe data feature tersebut untuk melakukan feature encoding.

Pre-processing (Merging Dataset)

Setelah membersihkan masing - masing dataset, kami me-**aggregating** data dan **menggabungkan** ke dataset **Application_Train & Test** berdasarkan **SK_ID_CURR** karena data yang ingin diuji adalah data Application_Train, maka dataset yang masuk adalah yang sesuai dengan SK_ID_CURR milik Application_Train.

Setelah digabungkan, kami melakukan **preprocessing ulang** terlihat beberapa **feature null value (di atas 40%)**, namun disini kami isi dengan **nilai 0** karena kemungkinan beberapa customer memang tidak memiliki riwayat transaksi di bureau, previous application, posh cash, ataupun credit card sehingga value pada feature - feature didataset tersebut null. Lalu yang **dibawah 40%** kami isi dengan **median** karena robust dan tidak terlalu mengubah keseluruhan data.

Untuk **handling outliers dan skewness** kami lakukan metode **yeo-johnson** karena metode tersebut dapat **handling** skew **positif** dan **negatif** serta melakukan **standardization** agar **skala** setiap feature **sama rata.**

```
Total rows in dataset: 307511

Null Value Analysis:
                                Total_Null  Null_Percentage
CCB_DRAWINGS_OTHER_PERCENTAGE_MAX     250458            81.45
CCB_DRAWINGS_OTHER_PERCENTAGE_MIN     250458            81.45
CCB_DRAWINGS_OTHER_PERCENTAGE_MEAN    250458            81.45
CCB_AVG_DRAWING_MIN                   250457            81.45
CCB_DRAWINGS_ATM_PERCENTAGE_MAX       250458            81.45
CCB_DRAWINGS_ATM_PERCENTAGE_MEAN      250458            81.45
CCB_DRAWINGS_POS_PERCENTAGE_MIN       250458            81.45
CCB_DRAWINGS_ATM_PERCENTAGE_MIN       250458            81.45
CCB_DRAWINGS_POS_PERCENTAGE_MAX       250458            81.45
CCB_AVG_DRAWING_MEAN                  250458            81.45
CCB_AVG_DRAWING_MAX                   250457            81.45
CCB_DRAWINGS_POS_PERCENTAGE_MEAN      250458            81.45
CCB_LIMIT_RATIO_MEAN                  224308            72.94
PREV_DAYS_TERMINATION_MIN             223570            72.70
PREV_DAYS_TERMINATION_MAX             223570            72.70
PREV_DAYS_TERMINATION_MEAN            223570            72.70
PREV_NAME_YIELD_GROUP_NUNIQUE         223570            72.70
PREV_SELLERPLACE_AREA_MEAN            223570            72.70
PREV_SELLERPLACE_AREA_NUNIQUE         223570            72.70
POS_SK_DPD_DEF_MEAN                   223570            72.70
PREV_AMT_DOWN_PAYMENT_MIN             223570            72.70
...
BUR_DAYS_CREDIT_UPDATE_MAX             44020            14.31
BUR_DAYS_CREDIT_UPDATE_MIN             44020            14.31
BUR_DAYS_ENDDATE_FACT_MEAN             44020            14.31
BUR_CREDIT_DAY_OVERDUE_MAX             44020            14.31
```

Modelling Experiments

Pada eksperimen modelling ini, kami telah melakukan berbagai versi, yang antara lain:

- Eksperimen yang kami lakukan pertama kali adalah memilih **feature yang berkorelasi** saja dan memilih **salah satu feature yang redundan** dengan tidak melibatkan Pos Cash Balance dan Installment karena d**ata terlalu banyak** sehingga **komputer kami tidak kuat** untuk menjalankan model.
- Karena hasil pertama tidak memuaskan, kami mencoba untuk **melakukan hyperparameter tuning dan sedikit preprocessing ulang**. namun, **komputer terlalu lama** untuk **menjalankan hyperparameter tuning** bahkan sering terjadi crash pada kernel yang mengharuskan untuk restart dan hasil tidak memuaskan.
- Setelah tidak ditemukan kemajuan score yang baik, kami memutuskan untuk melakukan preprocessing ulang. Tahapan preprocessing ini adalah apa yang telah kami sebutkan pada slide sebelumnya. Kemudian kami mencoba untuk melakukan modelling pada **Logistic Regression, Decision Tree, Random Forest, AdaBoost, dan XGBoost** karena terdapat outliers dan sering overfitting. Hasil menunjukkan Logistic Regression yang best fit dan kami menamai hasil ini sebagai model **v1.1**.
- Kemudian, kami ingin meningkatkan score metrics. Percobaan kami lakukan seperti **v1.2** kami lakukan **drop null value di atas 40%** (yang sebelumnya di isi dengan angka 0) dan **hasil tidak terlalu berbeda signifikan**, namun mengalami penurunan kurang lebih 0.01 dibandingkan versi v1.1.
- Pada **version v1.3** kami lakukan **handling outliers** dan hasil juga tidak menunjukan perubahan yang signifikan pada v1.2. Kami tidak terlalu fokus pada hyperparameter tuning karena data terlalu besar meskipun sudah di tuning dan sangat berat yang menyebabkan sering kali restart kernell.
- Karena keterbatasan waktu, kami memutuskan untuk menggunakan score pada version v1.1. yang telah kami lakukan.

Modelling Experiments

Pada modelling final ini, kami mempersiapkan data sebagai berikut:

- Membagi data **80% untuk training** dan **20% untuk testing** karena kelas pada label **imbalance ekstrim** sehingga model bisa semakin banyak mempelajari data.
- Ratio perbandingan **imbalance class adalah 92:8** dan agar **model tidak bias terhadap kelas majority** serta **tidak kehilangan banyak data penting**, dilakukan ***Synthethic Minority Oversampling Technique*** agar data minoritas dibuat baru secara sintetis sehingga tidak menduplikasi data dari minoritas.
- Model metrics yang kami gunakan adalah **ROC AUC** dan **Recall**. Alasan kami menggunakan dua metrics tersebut antara lain:
    - **ROC AUC** mampu membedakan kelas positif dan negatif sehingga lebih robust terhadap class imbalance.
    - **Recall** mampu menekan angka *False Negative* sehingga kesalahan prediksi ketika model memprediksi seseorang tidak default tapi ternyata default itu bisa diminimalisir karena **kehilangan potensial customer lebih baik daripada lolosnya customer yang pada akhirnya tidak mampu membayar pinjaman.**
- Menggunakan **Select KBest** untuk memilih jumlah feature yang diuji. Dari 339 features, kbest menunjukan bahwa semua feature penting sehingga model tetap menggunakan 339 features.
- Menguji pada algoritma Logistic Regression, Decision Tree, Random Forest, Ada Boost, dan XGBoost karena data yang diuji terdapat outliers dan hasil sering overfitting sehingga penggunaan seperti Random Forest dapat robust terhadap outliers. Kami tidak menguji menggunakan KNN, Support Vector Machine (SVM), Light GBM, dan lainnya karena keterbatasan waktu dan lama waktu untuk running model.
- Hyperparameter tuning tidak terlalu difokuskan karena lamanya waktu untuk tuning terkhusus pada bagian Decision Tree, Random Forest, dan Ada Boost.

Modelling Experiments

Logistic Regression	Train	Test
Recall	0.72	0.66
ROC AUC	0.77	0.74

Decision Tree	Train	Test
Recall	1.00	0.74
ROC AUC	1.00	0.50

Random Forest	Train	Test
Recall	0.99	0.72
ROC AUC	0.99	0.69

Ada Boost	Train	Test
Recall	0.88	0.98
ROC AUC	0.95	0.62

XGBoost	Train	Test
Recall	0.92	0.93
ROC AUC	0.99	0.38

Dari 5 algoritma yang telah dijalankan, berikut adalah penjelasan hasil:

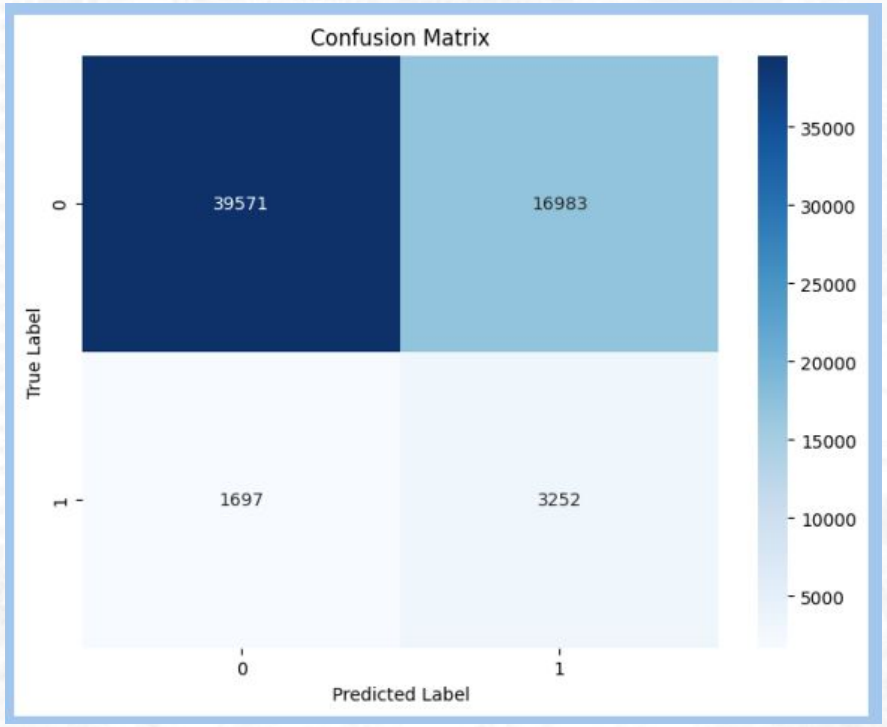
- Pada **Logistic Regression** telah dilakukan regularization Ridge untuk mengurangi kompleksitas model (variasi) dan menangani multikolinearitas dan score menunjukan **ROC AUC sebesar 0.74 dan recall sebesar 0.66** dengan **waktu tuning 1m 29.7s**.
- Pada **Decision Tree**, komputer kami **tidak mampu** untuk **running hyperparameter tuning** sehingga hasil cukup rendah dengan angka **ROC AUC 0.5** dan **recall 0.74**. Melihat selisih yang cukup jauh dengan data train menunjukan model masih overfitting dengan waktu running model 1m 31.1s.
- Random Forest berhasil dilakukan tuning meskipun memakan waktu 97m 36.7s dimana tuning mencari parameter terbaik dan parameter terpilih adalah jumlah pohon sebanyak 166, min_samples_split 7, min_samples_leaf 2, max_features log2, max_depth 74, dan handling ketidakseimbangan kelas menggunakan class_weight balanced. Hasil tidak menunjukan overfitting.
- Selanjutnya menggunakan AdaBoost dengan waktu tuning 72m 27.5s dengan parameter terbaik yaitu n_estimator 100, learning_rate 0.6, algorithm SAMME.R menghasilkan score ROC AUC 0.62 dan Recall yang tinggi yaitu 0.98 namun masih menunjukan overfitting.
- Dan yang terakhir yaitu XGBoost dengan lama tuning 1m 56.9s dan parameter terbaik yaitu subsample 0.85, scale_pos_weight 4, reg_lambda 0.3, reg_alpha 0, n_estimator 100, min_child_weight 5, max_depth 4, learning_rate 0.05, gamma 0.3, dan colsample_bytree 0.75, menghasilkan score recall yang sangat baik dan tidak overfitting yaitu 0.93, namun score ROC AUC yang sangat overfitting.

Modelling Experiments

Dari hasil 5 algoritma yang dijalankan, kami memutuskan untuk memilih logistic regression. Hal yang menjadi pertimbangan kami antara lain:

- Fokus utama kami adalah **metrics ROC AUC** dan metrics tersebut dilogistic regression terlihat best fit dibandingkan metrics yang lain.
- Meskipun **adaboost best fit pada recall (0.93)**, namun **ROC AUC** yang **sangat rendah** menyebabkan model terlalu **bias** terhadap kelas **positif**.
- Dari **confussion matrix** yang dibuat, kesalahan prediksi dimana model memprediksi orang itu tidak default tapi actualnya default (***False Negative***) itu hanya **2%** saja sehingga kerugian dapat ditekan.

Logistic Regression		
	Train	Test
Recall	0.72	0.66
ROC AUC	0.77	0.74



STAGE3 - Executive Summary & Recommendation

Default:

- Penilaian score perlu memberikan bobot lebih pada **AMT_GOODS_PRICE, EXT_SOURCE_MEAN, DEBT_TO_INCOME,** dan **AMT_INCOME_RATIO** karena hal ini penting dalam memprediksi seseorang gagal bayar atau tidak.
- Mengembangkan **risk-based pricing** untuk menentukan **suku bunga** berdasarkan customer risk.
- Melakukan **monitoring** untuk melihat default rate dan profit margins untuk setiap 4 bulan.

saran saat proses monitoring:

- Menerapkan **sistem auto reminder** melalui email, WhatsApp, dan notifikasi di smartphone.

Rekomendasi setelah 4 bulan pemantauan:

- Jika Tingkat **Default Rate Meningkat**:
  - Tetapkan batas kredit sesuai dengan kemampuan pelanggan.
  - Kurangi aktivitas pemberian pinjaman kepada pelanggan dengan risiko tinggi.
- Jika Tingkat **Default Rate Menurun**:
  - Tawarkan produk keuangan lain kepada pelanggan yang selalu membayar tepat waktu.
  - Berikan promosi, seperti pengajuan pinjaman jangka panjang dengan manfaat tambahan, seperti pembebasan biaya tahunan.

Pembagian Tugas

Stage	Tugas	Penanggung Jawab	Keterangan
Stage 0	Problem Statement	Semua anggota t...	Masing - masing anggota tim mengerjakan tugasnya masing - masing dan brainstorming dari hasil analisis dan ide yang didapat
	Goal		
	Objective		
	Role		
	Business Metrics		
	Laporan PPT		
	Presentasi dengan mentor	Andiny Lusy S.	
Stage 1	Statistics Descriptive	Aldi Vibriani	
	Univariate Analysis	Andiny Lusy S.	
	Multivariate Analysis	Aldi Vibriani	
	Visualization for Business Insight	Aldi Vibriani	Application_train & test
		Alfriando C. Vean	Application_train & test, bureau, installment, credit card balance, dan pos cash balance
		Andiny Lusy S.	Application_train & test, bureau, installment, credit card balance, dan pos cash balance
		David Yudha P.	Application_train & test
		Ramadhian E.	Application_train & test, previous application
		Reza Yulhansyah	Application_train & test, bureau, installment, credit card balance, dan pos cash balance
		Nur Alifah Z.	Application_train & test, bureau & bureau balance
	Upload Github	Aldi Vibriani	Ketua kelompok yang melakukan upload ke Github
	Laporan PPT	Aldi Vibriani	
		Alfriando C. Vean	
		Andiny Lusy S.	
		Ramadhian E.	
		Reza Yulhansyah	
	Presentasi dengan mentor	Aldi Vibriani	

Stage 2	Application Train & Test Data Preprocessing	Aldi Vibriani	Semua anggota tim melakukan data preprocessing pada application train and test sesuai dengan feature yang mereka dapatkan karena feature pada dataset sangat banyak.
		Alfriando C. Vean	
		Andiny Lusy S.	
		David Yudha P.	
		Nur Alifah Z.	
		Ramadhian E.	
		Reza Yulhansyah	
	Bureau & Bureau Balance Data Preprocessing	Andiny Lusy S.	Preprocessing dataset Bureau & Bureau Balance
		Nur Alifah Z.	Membantu kak Lusy preprocessing
		Reza Yulhansyah	Preprocessing ulang untuk scenario pemodelan
	Previous Application Preprocessing	Aldi Vibriani	Preprocessing dataset Previous Application
	Pos Cash Balance Preprocessing	Reza Yulhansyah	Preprocessing dataset Pos Cash Balance
	Credit Card Balance Preprocessing	Andiny Lusy S.	Preprocessing dataset Credit Card Balance
	Preprocessing Ulang untuk scenario di modelling baru	Aldi Vibriani	Melakukan preprocessing mandiri untuk model di scenario lain
	Preprocessing Ulang untuk scenario di modelling baru	David Yudha P.	Melakukan preprocessing mandiri untuk model di scenario lain
	Upload Github	Aldi Vibriani	Ketua kelompok yang melakukan upload ke Github
	Laporan	Semua anggota t...	Semua anggota tim melakukan pelaporan pada Stage 2
	Presentasi dengan mentor	Ramadhian E.	
Stage 3	Modelling Machine Learning	Alfriando C. Vean	Melakukan model machine learning pada Data Application Train dan Test.
		Reza Yulhansyah	
		Aldi Vibriani	Me-support Data Scientist Team dalam proses modelling dan keperluan data cleansing.
		Andiny Lusy S.	Me-support Data Scientist Team dalam proses modelling dan keperluan data cleansing.
		David Yudha P.	Me-support Data Scientist Team dalam proses modelling dan keperluan data cleansing.
		Nur Alifah Z.	Me-support Data Scientist Team dalam proses modelling dan keperluan data cleansing.
	Presentasi dengan mentor	Aldi Vibriani	

Stage 4	Laporan PPT	Andiny Lusy S.	Mengerjakan pembuatan laporan final project
		Aldi Vibriani	Membantu pembuatan laporan final project
		Alfriando C. Vean	Membantu pembuatan laporan final project
		Reza Yulhansyah	Membantu pembuatan laporan final project
	Laporan Final Project	Aldi Vibriani	Mengerjakan pembuatan laporan final project
		Andiny Lusy S.	Membantu pembuatan laporan final project
	Notulen stage 0	Ramadhian E.	
	Notulen stage 1	Alfriando C. Vean	
		Reza Yulhansyah	
	Notulen stage 2	Andiny Lusy S.	
		Nur Alifah Z.	
	Notulen stage 3	Aldi Vibriani	
		Nur Alifah Z.	
	Notulen stage 4	Aldi Vibriani	
	Presentasi dengan mentor	Aldi Vibriani	Simulasi Presentasi dengan mentor untuk Final Project
		Andiny Lusy S.	
		Alfriando C. Vean	
		David Yudha P.	
		Reza Yulhansyah	

Link spreadsheet Kontribusi anggota Octagram: [Click here]

Rakamin Academy
School for Career Acceleration

Terima Kasih