APRIL 2024

FINAL TASK

Credit Loan Prediction

Latar Belakang Masalah

Membangun model yang dapat memprediksi credit risk menggunakan dataset yang disediakan oleh company yang terdiri dari data pinjaman yang diterima dan yang ditolak. Diperlukan untuk mempersiapkan media visual dalam mempresentasikan solusi ke klien.

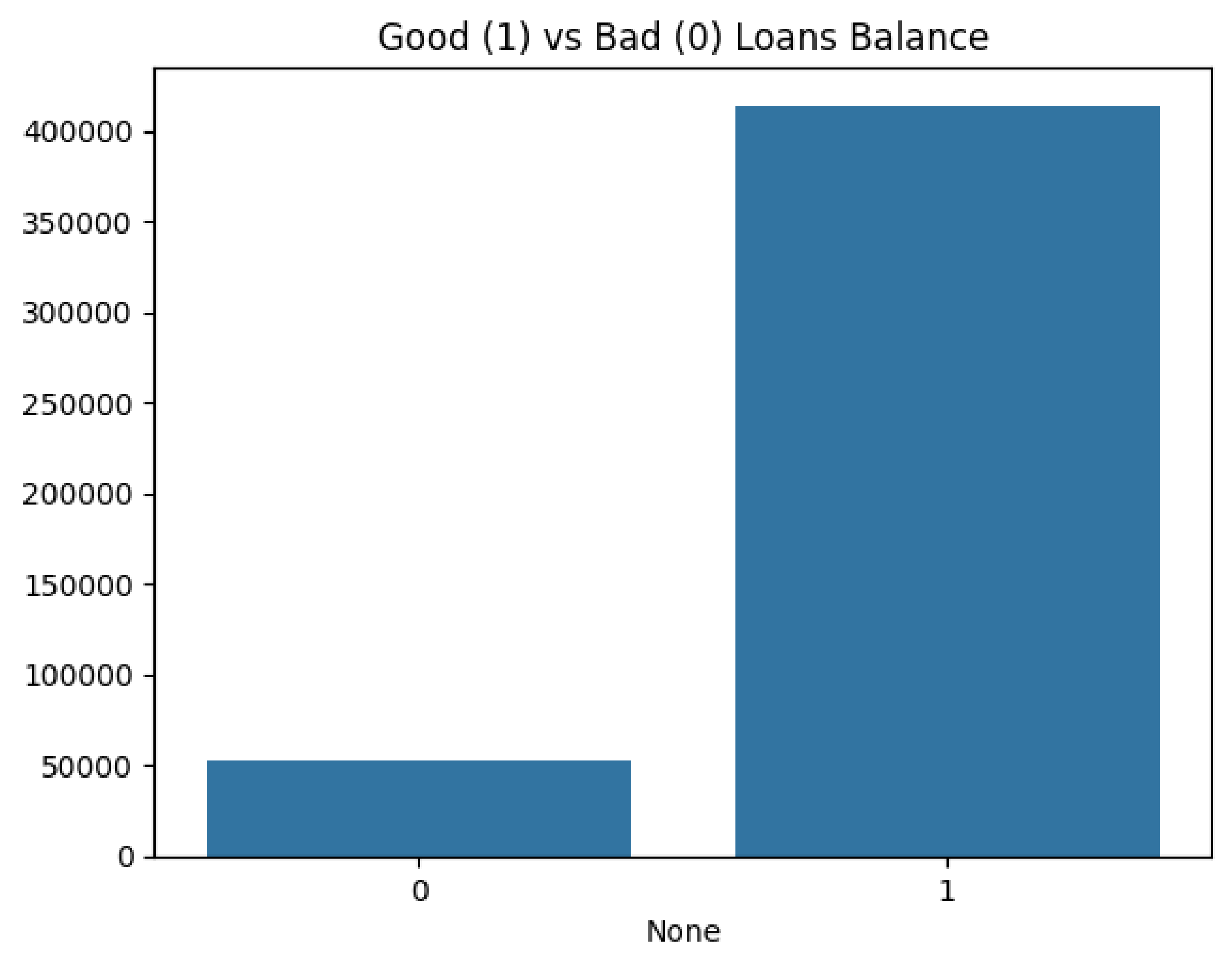
Exploratory Data Analysis

```
Data columns (total 75 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   Unnamed: 0                   466285 non-null  int64
 1   id                           466285 non-null  int64
 2   member_id                    466285 non-null  int64
 3   loan_amnt                    466285 non-null  int64
 4   funded_amnt                  466285 non-null  int64
 5   funded_amnt_inv              466285 non-null  float64
 6   term                         466285 non-null  object
 7   int_rate                     466285 non-null  float64
 8   installment                  466285 non-null  float64
 9   grade                        466285 non-null  object
 10  sub_grade                    466285 non-null  object
 11  emp_title                    438697 non-null  object
 12  emp_length                   445277 non-null  object
 13  home_ownership               466285 non-null  object
 14  annual_inc                   466281 non-null  float64
 15  verification_status          466285 non-null  object
 16  issue_d                      466285 non-null  object
 17  loan_status                  466285 non-null  object
 18  pymnt_plan                   466285 non-null  object
 19  url                          466285 non-null  object
 20  desc                         125983 non-null  object
 21  purpose                      466285 non-null  object
 22  title                        466265 non-null  object
 23  zip_code                     466285 non-null  object
 24  addr_state                   466285 non-null  object
 25  dti                          466285 non-null  float64
 26  delinq_2yrs                  466256 non-null  float64
 27  earliest_cr_line             466256 non-null  object
 28  inq_last_6mths               466256 non-null  float64
 29  mths_since_last_delinq       215934 non-null  float64
 30  mths_since_last_record       62638 non-null   float64
 31  open_acc                     466256 non-null  float64
 32  pub_rec                      466256 non-null  float64
 33  revol_bal                    466285 non-null  int64
```

- Terdapat 466285 rows dan 75 features
- Terdapat type data yang keliru
- Terdapat nilai data yang hilang
- Tidak ada outliers

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	int_rate	installment	annual_inc	dti	...
count	466285.000000	4.662850e+05	4.662850e+05	466285.000000	466285.000000	466285.000000	466285.000000	466285.000000	4.662810e+05	466285.000000	...
mean	233142.000000	1.307973e+07	1.459766e+07	14317.277577	14291.801044	14222.329888	13.829236	432.061201	7.327738e+04	17.218758	...
std	134605.029472	1.089371e+07	1.168237e+07	8286.509164	8274.371300	8297.637788	4.357587	243.485550	5.496357e+04	7.851121	...
min	0.000000	5.473400e+04	7.047300e+04	500.000000	500.000000	0.000000	5.420000	15.670000	1.896000e+03	0.000000	...
25%	116571.000000	3.639987e+06	4.379705e+06	8000.000000	8000.000000	8000.000000	10.990000	256.690000	4.500000e+04	11.360000	...
50%	233142.000000	1.010790e+07	1.194108e+07	12000.000000	12000.000000	12000.000000	13.660000	379.890000	6.300000e+04	16.870000	...
75%	349713.000000	2.073121e+07	2.300154e+07	20000.000000	20000.000000	19950.000000	16.490000	566.580000	8.896000e+04	22.780000	...
max	466284.000000	3.809811e+07	4.086083e+07	35000.000000	35000.000000	35000.000000	26.060000	1409.990000	7.500000e+06	39.990000	...

Exploratory Data Analysis

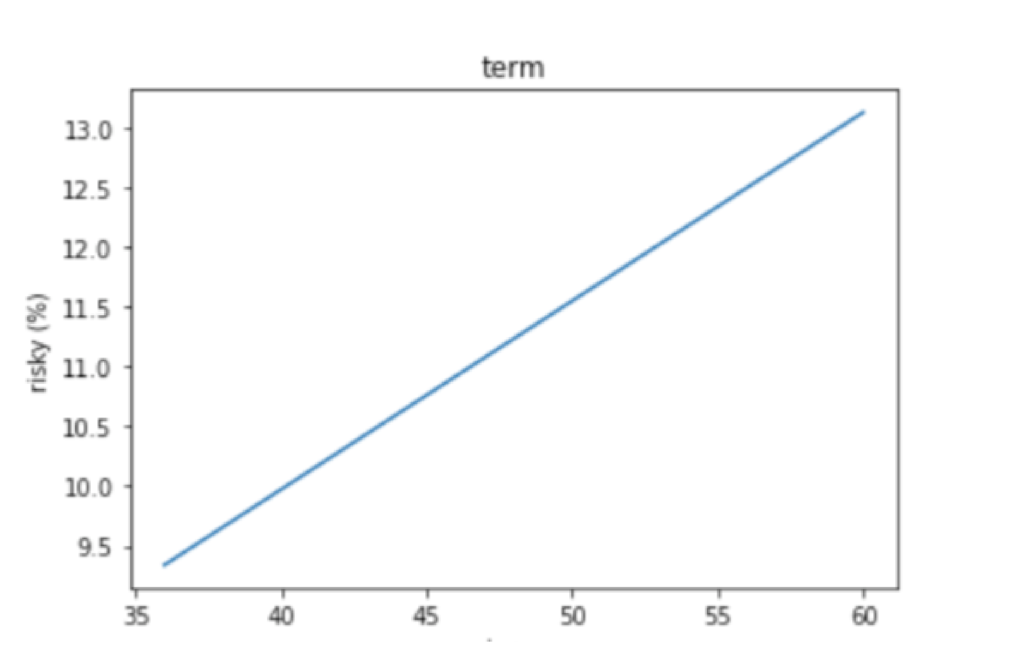


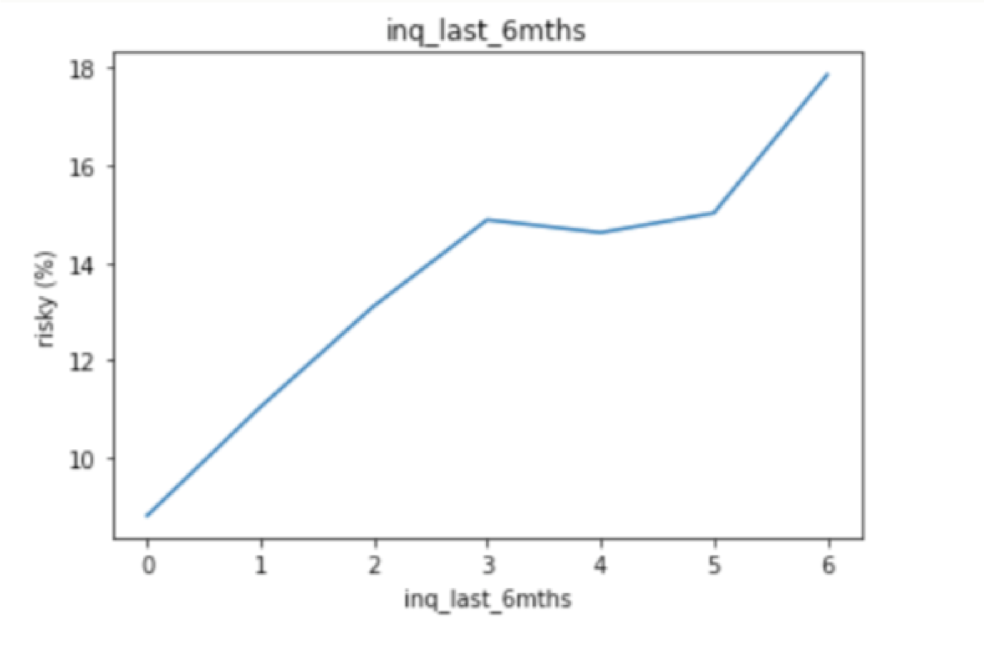
Pada data target yang digunakan yaitu ‘loan_status’ yang akan dibagi menjadi 2 kategori yaitu ‘good’ dan ‘bad’ berdasarkan value unique yang sebelumnya sebagai berikut.

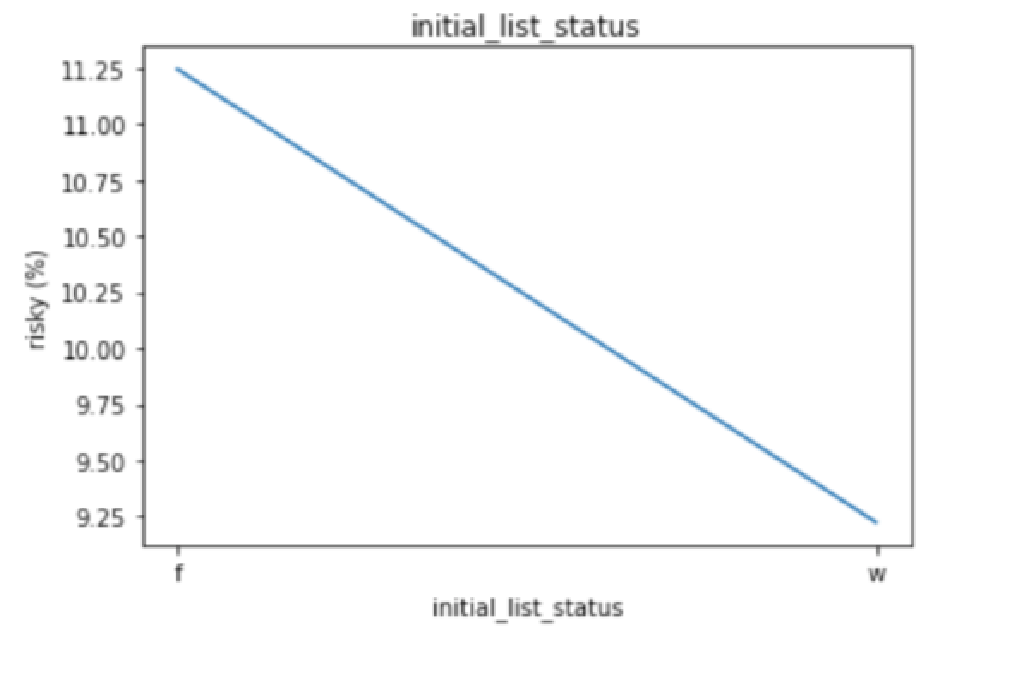
- good loans = ['Current', 'Fully Paid', 'In Grace Period', 'Does not meet the credit policy. Status:Fully Paid']
- bad loans = ['Charged Off', 'Late (31-120 days)', 'Late (16-30 days)', 'Default', 'Does not meet the credit policy. Status:Charged Off']

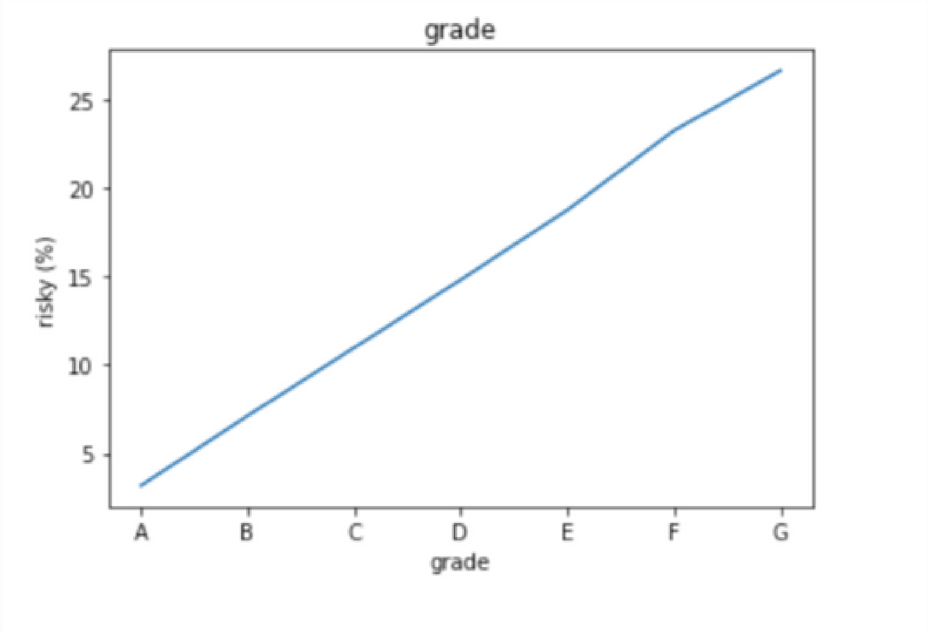
Namun perbandingan antara ‘good’ dan ‘bad’ terdapat imbalance data sehingga data yang akan dilakukan prediksi menjadi kurang realistis. Maka dari itu pada kasus ini diatasi dengan cara oversampling.

Exploratory Data Analysis



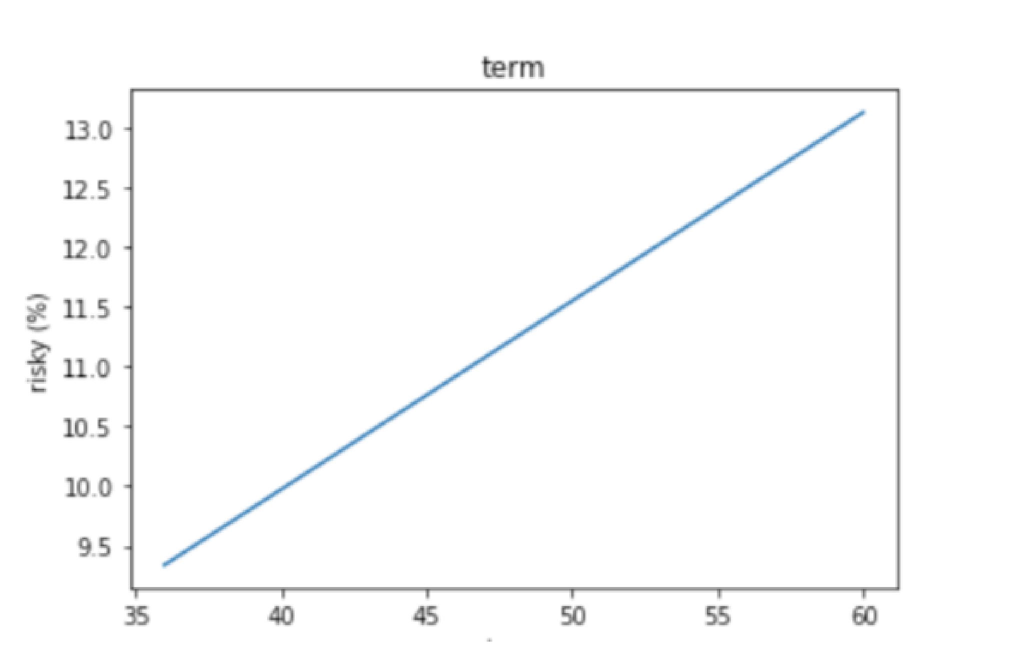


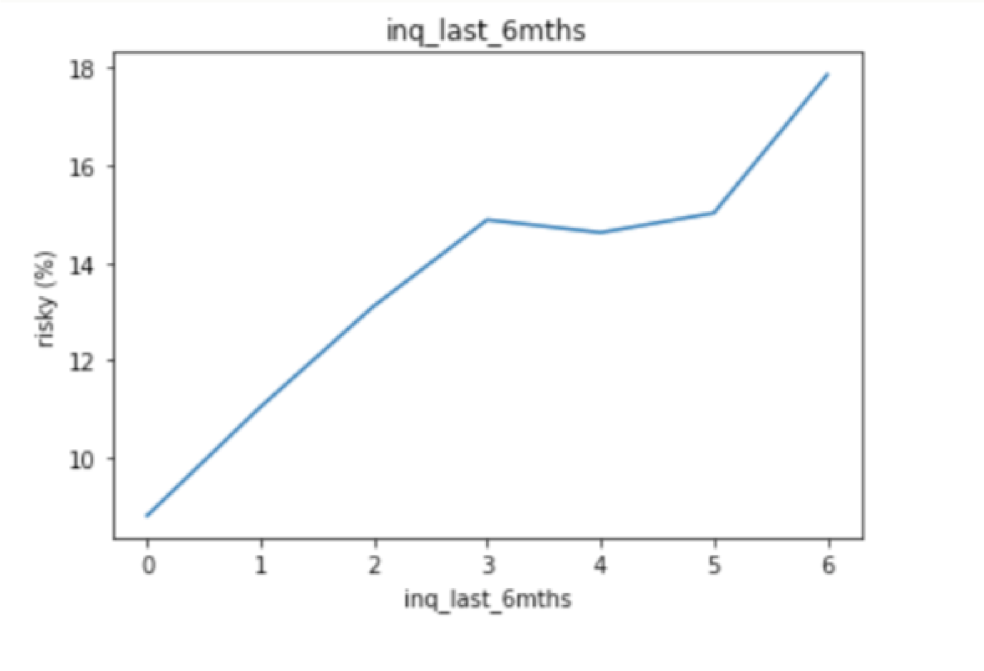


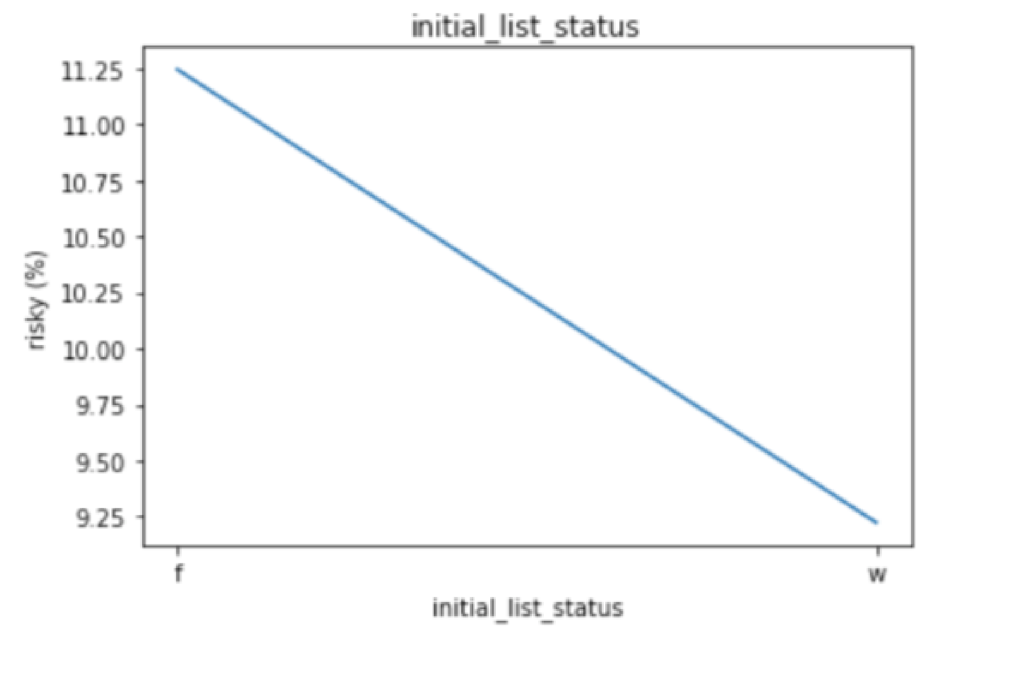


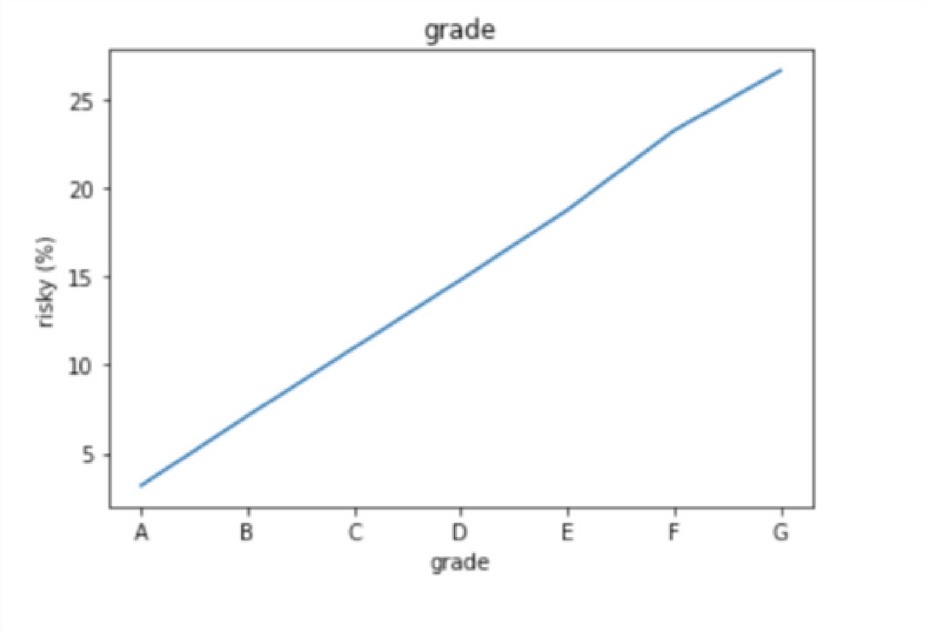
- term: have low risk at term 36 and high risk at term 60
- inq_last_6mths: there is an increased risk associated with this.
- initial_list_status: have high risk in f and low risk in w
- grade: there is an increased risk associated with this.

Exploratory Data Analysis



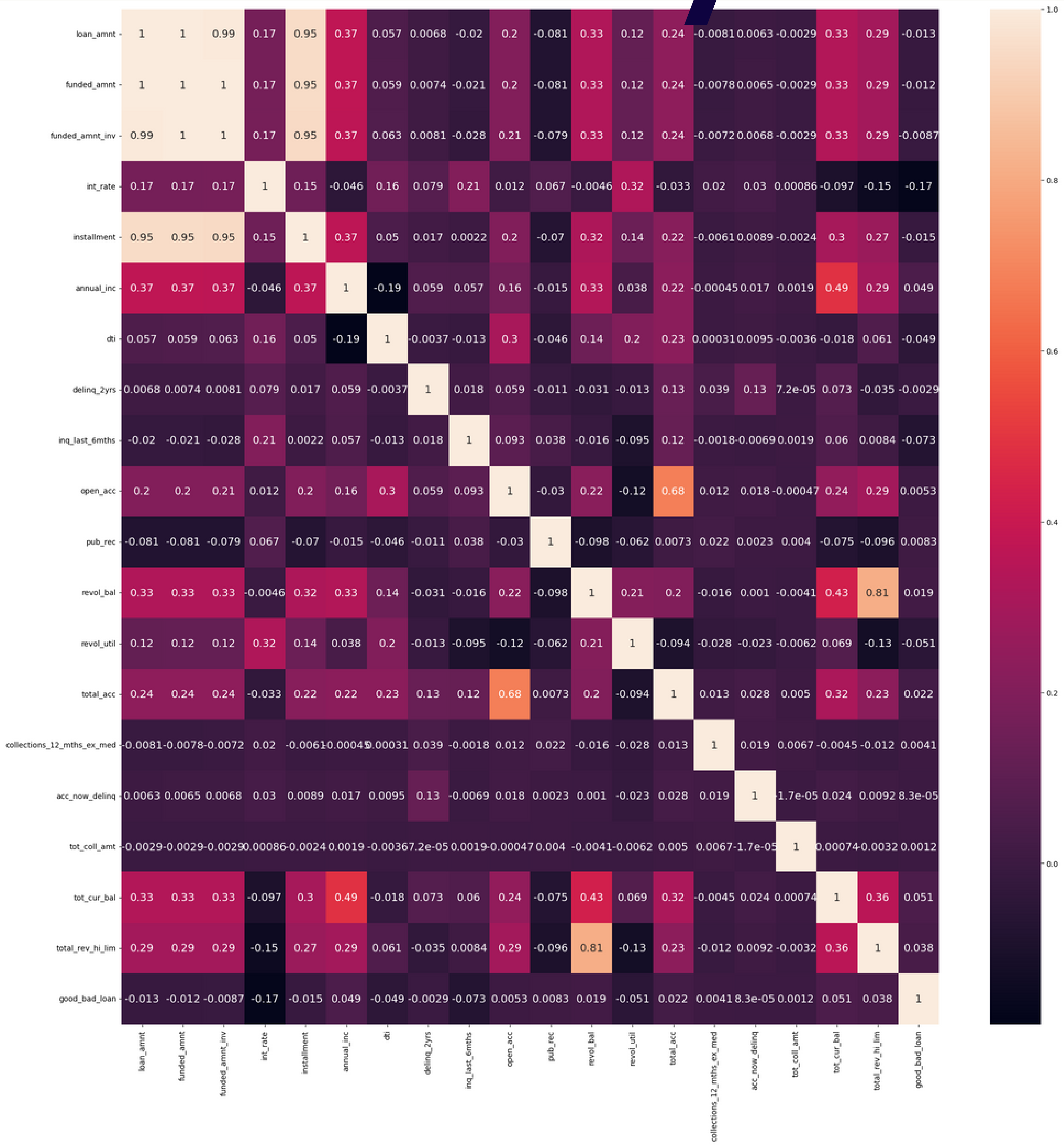






- term: have low risk at term 36 and high risk at term 60
- inq_last_6mths: there is an increased risk associated with this.
- initial_list_status: have high risk in f and low risk in w
- grade: there is an increased risk associated with this.

Exploratory Data Analysis



- tloan_amnt, funded_amnt, funded_amnt_inv memiliki korelasi yang mirip dengan kolom lainnya sehingga kolom-kolom tersebut cenderung memiliki kemiripan data

Pre-Processing

```
desc                            0.729815
mths_since_last_delinq          0.536906
mths_since_last_record          0.865666
mths_since_last_major_derog     0.787739
annual_inc_joint                1.000000
dti_joint                       1.000000
verification_status_joint       1.000000
open_acc_6m                     1.000000
open_il_6m                      1.000000
open_il_12m                     1.000000
open_il_24m                     1.000000
mths_since_rcnt_il              1.000000
total_bal_il                    1.000000
il_util                         1.000000
open_rv_12m                     1.000000
open_rv_24m                     1.000000
max_bal_bc                      1.000000
all_util                        1.000000
inq_fi                          1.000000
total_cu_tl                     1.000000
inq_last_12m                    1.000000
dtype: float64
```

Data Cleaning and Check Duplicate

Pada data ini memiliki beberapa missing values sehingga pada kasus ini dilakukan pengecekan missing values berdasarkan:

- Drop column 'Unnamed: 0' which is a copy of an index.
- Drop the columns having > 50% missing values. (columns with 0 unique value are also columns that have 100% missing value)
- Drop column 'application_type' and 'policy_code' (it only have 1 unique value).
- Drop identifier columns: id, member_id, title, emp_title, url, zip_code, desc, policy_code (it can not be used in building model).
- Drop sub_grade, it contains the same information as the grade columns.

Pre-Processing

#	Column	Non-Null Count	Dtype
0	loan_amnt	396009 non-null	int64
1	term	396009 non-null	int64
2	int_rate	396009 non-null	float64
3	installment	396009 non-null	float64
4	grade	396009 non-null	object
5	emp_length	396009 non-null	int64
6	home_ownership	396009 non-null	object
7	annual_inc	396009 non-null	float64
8	verification_status	396009 non-null	object
9	purpose	396009 non-null	object
10	addr_state	396009 non-null	object
11	dti	396009 non-null	float64
12	delinq_2yrs	396009 non-null	float64
13	inq_last_6mths	396009 non-null	float64
14	open_acc	396009 non-null	float64
15	pub_rec	396009 non-null	float64
16	revol_bal	396009 non-null	int64
17	total_acc	396009 non-null	float64
18	initial_list_status	396009 non-null	object
19	collections_12_mths_ex_med	396009 non-null	float64
20	acc_now_delinq	396009 non-null	float64
21	tot_coll_amt	396009 non-null	float64
22	tot_cur_bal	396009 non-null	float64
23	total_rev_hi_lim	396009 non-null	float64
24	good_bad_loan	396009 non-null	int64
25	mths_since_earliest_cr_line_date	396009 non-null	float64
26	mths_since_last_credit_pull_d	396009 non-null	float64

Feature Selection

Pada kasus ini hanya memilih kolom-kolom tertentu saja yang akan dijadikan fitur independen pada pemodelan nantinya

Pre-Processing

```
# Convert categorical columns with One Hot Encoding
from sklearn.preprocessing import OneHotEncoder
cat_cols = [col for col in loan_data.select_dtypes(include='object').columns.tolist()]
onehot_cols = pd.get_dummies(loan_data[cat_cols], drop_first=True)
```

Encoding Categorical to Numerical

Pada kolom yang bersifat kategorikal diubah menjadi numerik dahulu sebelum melanjutkan ke tahap pemodelan. Pada proses encoding ini dilakukan dengan cara One-Hot Encoding.

```
from sklearn.preprocessing import StandardScaler

num_cols = [col for col in loan_data.columns.tolist() if col not in cat_cols + ['good_bad_loan']]
ss = StandardScaler()
std_cols = pd.DataFrame(ss.fit_transform(loan_data[num_cols]), columns=num_cols)
```

Scaling data dengan StandardScaler

Pre-Processing

Scaling data dengan StandardScaler

Before

loan_amnt	term	int_rate	installment	emp_length	annual_inc	dti	delinq_2yrs	inq_last_6mths	open_acc	pub_rec	revol_bal	total_acc
27050	36	10.99	885.46	10	55000.0	22.87	0.0	0.0	14.0	0.0	36638	27.0
9750	36	13.98	333.14	1	26000.0	25.12	0.0	0.0	12.0	0.0	7967	28.0
12000	36	6.62	368.45	10	105000.0	14.05	0.0	1.0	12.0	0.0	13168	22.0
12000	36	13.53	407.40	10	40000.0	16.94	0.0	0.0	7.0	2.0	5572	32.0
15000	36	8.90	476.30	2	63000.0	16.51	0.0	0.0	8.0	0.0	11431	29.0
...	...	...	...	...	...	...	...	...	...	...	...	...
18400	60	14.47	432.64	4	110000.0	19.85	0.0	2.0	18.0	0.0	23208	36.0
22000	6											
20700	6											
2000	3											
10000	3											

After

loan_amnt	term	int_rate	installment	emp_length	annual_inc	dti	delinq_2yrs	inq_last_6mths	open_acc	pub_rec	revol_bal
1.482170	-0.628523	-0.699230	1.805874	1.103741	-0.348864	0.652631	-0.369424	-0.747992	0.508914	-0.332097	0.958438
-0.600856	-0.628523	-0.016330	-0.459863	-1.286462	-0.879727	0.938245	-0.369424	-0.747992	0.111068	-0.332097	-0.410679
-0.329942	-0.628523	-1.697314	-0.315014	1.103741	0.566416	-0.466979	-0.369424	0.217036	0.111068	-0.332097	-0.162317
-0.329942	-0.628523	-0.119108	-0.155232	1.103741	-0.623449	-0.100122	-0.369424	-0.747992	-0.883548	3.345057	-0.525047
0.031276	-0.628523	-1.176575	0.127411	-1.020884	-0.202420	-0.154707	-0.369424	-0.747992	-0.684625	-0.332097	-0.245264
...	...	...	...	...	...	...	...	...	...	...	...
0.440657	1.591031	0.095583	-0.051692	-0.489728	0.657944	0.269273	-0.369424	1.182065	1.304607	-0.332097	0.317120
0.874119	1.591031	1.351753	0.563066	1.103741	0.072165	0.091557	-0.369424	4.077152	1.304607	1.506480	0.079789
0.717591	1.591031	0.671137	0.283459	0.307007	-0.513615	1.005524	-0.369424	1.182065	1.304607	-0.332097	-0.471755
-1.534004	-0.628523	-1.404969	-1.569718	-0.755306	0.163693	-1.566277	3.226411	0.217036	1.901376	-0.332097	-0.246553
-0.570755	-0.628523	1.175889	-0.318583	1.103741	-0.513615	0.641206	0.829187	-0.747992	-1.082471	-0.332097	-0.250326

Pre-Processing

Handling imbalance class

```
from imblearn.over_sampling import RandomOverSampler

ros = RandomOverSampler()
X_train_ros, y_train_ros = ros.fit_resample(X_train, y_train)

#check value counts before and after oversampling
print('Before OverSampling:\n{}'.format(y_train.value_counts()))
print('\nAfter OverSampling:\n{}'.format(y_train_ros.value_counts()))
```

```
Before OverSampling:
1    283815
0     32992
Name: good_bad_loan, dtype: int64

After OverSampling:
1    283815
0    283815
Name: good_bad_loan, dtype: int64
```

Imbalance pada data target dilakukan oversampling dengan menggunakan RandomOverSampler

Pre-Processing

Train Test Split

```
#spliting data into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42,stratify=y)
```

Pada pemodelan yang dipersiapkan dilakukan train test split data dengan porsi train 80 dan test 20

Modelling

Pada pemodelan dilakukan menggunakan 10 model diantaranya yaitu:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Ada Boost Classifier
- K Neighbors Classifier
- XGB Classifier
- LGBM Classifier
- GaussianNB
- QuadraticDiscriminantAnalysis
- MLPClassifier

Modelling Result

```
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000
Classification_Report:
              precision    recall  f1-score   support

    bad loan     0.2229    0.6182    0.3277      8248
   good loan     0.9441    0.7495    0.8356     70954

    accuracy                         0.7358     79202
   macro avg     0.5835    0.6839    0.5817     79202
weighted avg     0.8690    0.7358    0.7827     79202
```

Pada pemodelan yang telah dilakukan didapat bahwa model LGBM memiliki akurasi terbaik dimana Nilai akurasi rata-rata adalah 73,58% (recall pinjaman buruk = 61,82% dan recall pinjaman baik = 74,95%). Meskipun nilai akurasi ini masih belum tinggi, nilai ini sudah cukup tinggi karena dataset yang tidak seimbang. Recall adalah jumlah prediksi "positif" yang benar dibagi dengan total jumlah "positif". Ini berarti model ini berhasil mengidentifikasi 61,82% dari total pinjaman buruk dan berhasil mengidentifikasi 74,95% dari total pinjaman baik.

Terimakasih