Prediction of Home Credit Default Risk

HOME CREDIT

CISC-6080

Wanjia Song

1.Description

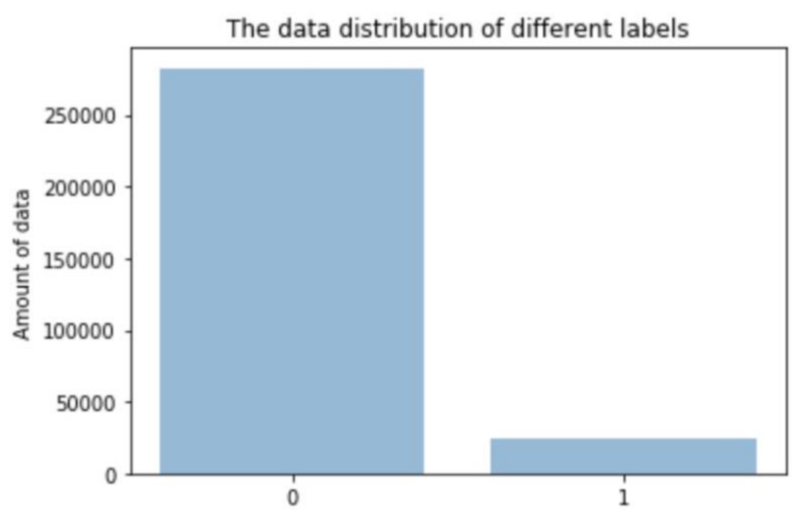
This project is to predict whether the person can repay the loan on time or not.
The data is provided by Home Credit from Kaggle competition. And the goal of the project is to ensure that clients capable of repayment are not rejected and that clients with potential loan problem are identified. There are four parts of this project: 1) Exploratory Data Analysis 2) Data Preprocessing 3) Modeling 4) Modeling turning 5) Esembling 6)Conclusion

2.Exploratory Data Analysis

Exploratory Data Analysis is a process where we calculate statistics and make figures to find trends, anomalies, patterns or relationships within the data.

1) Basic Description of data
   Our data has 307511 rows and 122 features. The features of the data mostly belong to four categories: Employment Status, Personal Asset, Personal Information and Credit Score.
   Employment Status: How many days employed, Occupation type, Organization Type.....
   Personal Asset: Apartment Avg, Living Apartment Avg...
   Personal Information: Age, Education, Family Status…
   Credit Score: EXT_SOURCE

2) The distribution of the Target Column
   The target is what we want to predict: **either a 0 for the loan was repaid on time, or a 1 indicating the client had payment difficulties.**



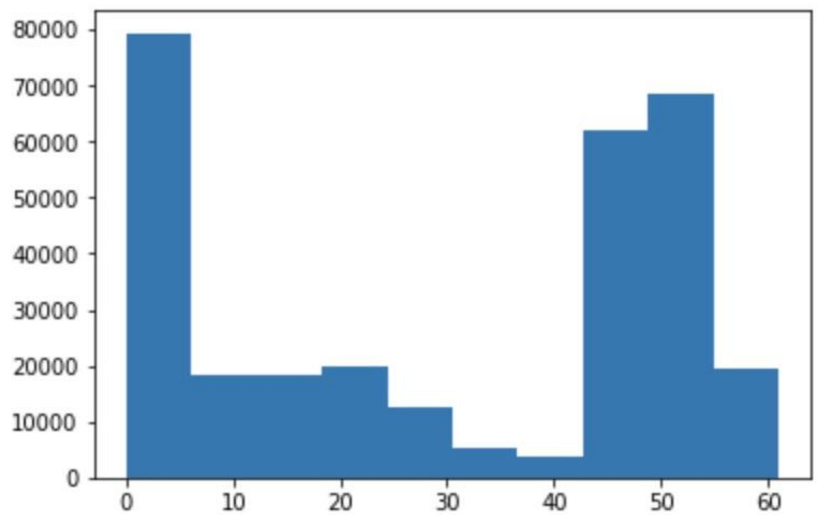
From the graph, we can see that the number of the data from class 0 is 282686, whereas the number of the data from class 1 is 24825. It is a really imbalanced dataset where the amount of minority data is only 10% of the majority data

3) Checking the missing values
For column level

	Missing value counts	Missing value percentage
COMMONAREA_MEDI	214865	0.698723
COMMONAREA_AVG	214865	0.698723
COMMONAREA_MODE	214865	0.698723
NONLIVINGAPARTMENTS_MODE	213514	0.694330
NONLIVINGAPARTMENTS_AVG	213514	0.694330
NONLIVINGAPARTMENTS_MEDI	213514	0.694330
FONDKAPREMONT_MODE	210295	0.683862
LIVINGAPARTMENTS_MODE	210199	0.683550
LIVINGAPARTMENTS_AVG	210199	0.683550
LIVINGAPARTMENTS_MEDI	210199	0.683550
FLOORSMIN_AVG	208642	0.678486
FLOORSMIN_MODE	208642	0.678486
FLOORSMIN_MEDI	208642	0.678486
YEARS_BUILD_MEDI	204488	0.664978
YEARS_BUILD_MODE	204488	0.664978
YEARS_BUILD_AVG	204488	0.664978
OWN_CAR_AGE	202929	0.659908
LANDAREA_MEDI	182590	0.593767
LANDAREA_MODE	182590	0.593767
LANDAREA_AVG	182590	0.593767
BASEMENTAREA_MEDI	179943	0.585160
BASEMENTAREA_AVG	179943	0.585160
BASEMENTAREA_MODE	179943	0.585160

We have checked the missing values and we can see that there are many features having large percentage of missing values. There are total 67 features have missing values.

For row level
There are still a large amount missing value percentage for each row. We can see that many rows have the missing value percentage larger than 45%

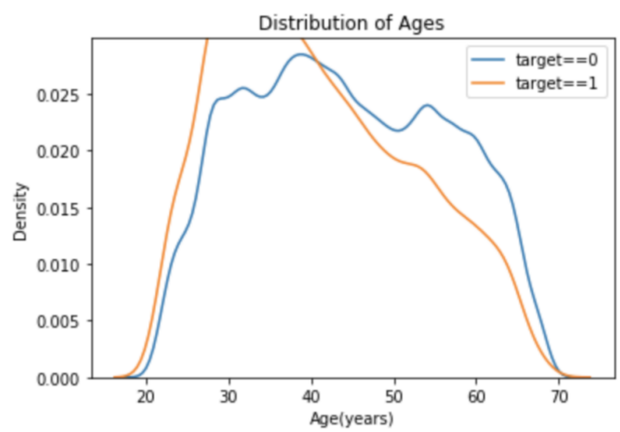


4) Correlation

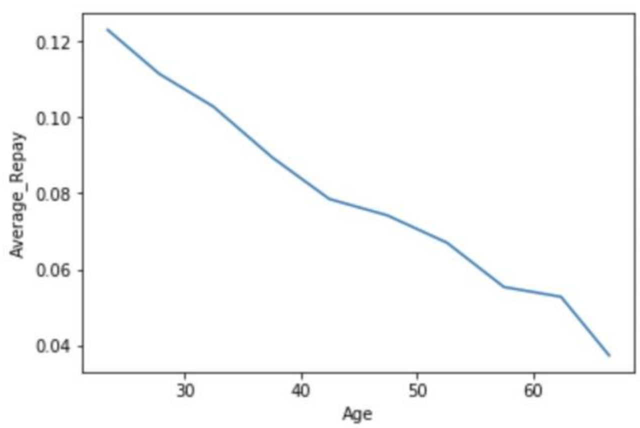
```
Most positive correlations: TARGET
DAYS_BIRTH                         0.078239
REGION_RATING_CLIENT_W_CITY        0.060893
REGION_RATING_CLIENT               0.058899
DAYS_LAST_PHONE_CHANGE             0.055218
DAYS_ID_PUBLISH                    0.051457
REG_CITY_NOT_WORK_CITY             0.050994
FLAG_EMP_PHONE                     0.045982
REG_CITY_NOT_LIVE_CITY             0.044395
FLAG_DOCUMENT_3                    0.044346
DAYS_REGISTRATION                  0.041975
OWN_CAR_AGE                        0.037612
LIVE_CITY_NOT_WORK_CITY            0.032518
DEF_30_CNT_SOCIAL_CIRCLE           0.032248
DEF_60_CNT_SOCIAL_CIRCLE           0.031276
Name: TARGET, dtype: float64

Most negative correlations: EXT_SOURCE_3
EXT_SOURCE_2                      -0.160472
EXT_SOURCE_1                      -0.155317
DAYS_EMPLOYED                     -0.044932
FLOORSMAX_AVG                     -0.044003
FLOORSMAX_MEDI                    -0.043768
FLOORSMAX_MODE                    -0.043226
AMT_GOODS_PRICE                   -0.039645
REGION_POPULATION_RELATIVE        -0.037227
ELEVATORS_AVG                     -0.034199
ELEVATORS_MEDI                    -0.033863
FLOORSMIN_AVG                     -0.033614
FLOORSMIN_MEDI                    -0.033394
LIVINGAREA_AVG                    -0.032997
LIVINGAREA_MEDI                   -0.032739
```

We observe the correlation between each feature and the label and just look at the most positive and negative correlations. We find that the most positive correlations is days_birth, the most negative correlation is ext_source_2 (which is a type of credit scoring)

5) The impact of age in repay



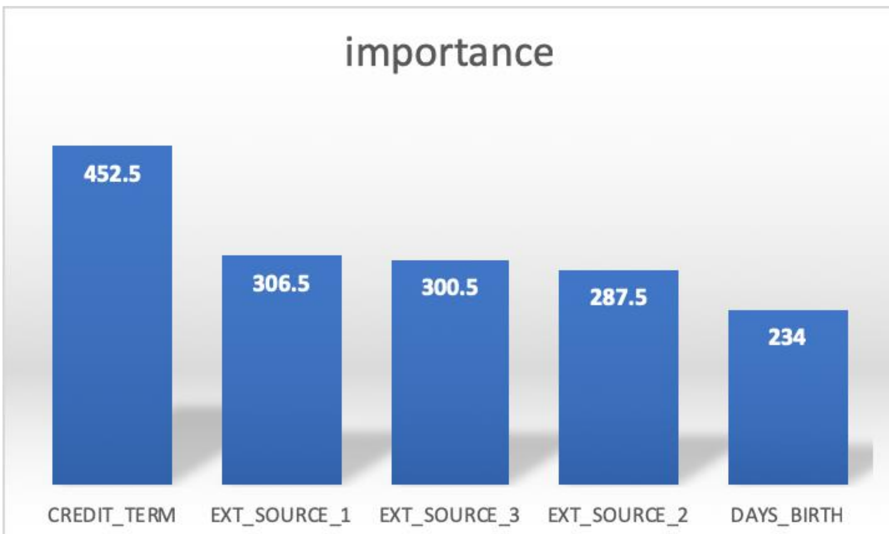
Here is the density distribute for different ages. We can see that for target=0, person tend to have a younger age, whereas for target=1, person tend to have an older age. We can have an assumption that young people may more possible to have a repay problem.



This is the graph shows the average of repay for different ages. We can still see that as the age increase, the repay tend to decrease to zero. (zero means they have no problem in repay)

6) Feature importance using lightbgm
Lightbgm is a type of gradient boosting method that uses tree-based algorithm. It has faster training speed and high efficiency. When there are missing values, it still can be implemented.

	feature	importance
196	CREDIT_TERM	452.5
27	EXT_SOURCE_1	306.5
29	EXT_SOURCE_3	300.5
28	EXT_SOURCE_2	287.5
9	DAYS_BIRTH	234.0



We use the feature importance in lightbgm. And we can find that the most important feature is credit_term, which indicates when payment is due for sales made on account. Ext_source is still important feature in this case.

3. Data Preprocessing

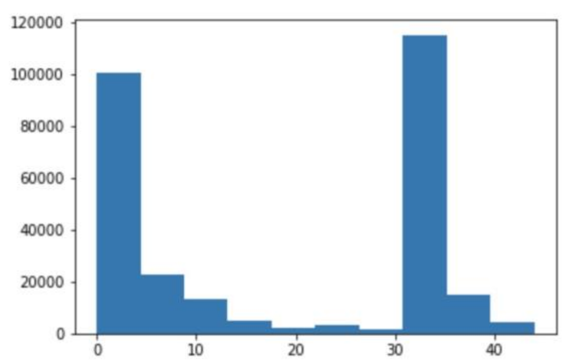
1) Delete the features with the highest missing value percentage

We decide to delete the data with the highest missing value percentage larger than 60%. And after doing this, we can get the missing data percentage as following.

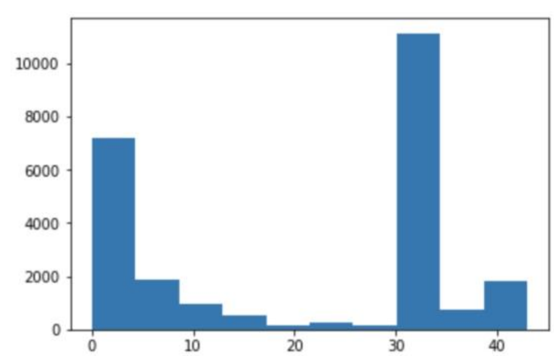
	Missing value counts	Missing value percentage
LANDAREA_AVG	182590	0.593767
LANDAREA_MODE	182590	0.593767
LANDAREA_MEDI	182590	0.593767
BASEMENTAREA_AVG	179943	0.585160
BASEMENTAREA_MODE	179943	0.585160
BASEMENTAREA_MEDI	179943	0.585160
EXT_SOURCE_1	173378	0.563811
NONLIVINGAREA_AVG	169682	0.551792
NONLIVINGAREA_MEDI	169682	0.551792
NONLIVINGAREA_MODE	169682	0.551792
ELEVATORS_MEDI	163891	0.532960
ELEVATORS_AVG	163891	0.532960
ELEVATORS_MODE	163891	0.532960
WALLSMATERIAL_MODE	156341	0.508408
APARTMENTS_MODE	156061	0.507497

2) Delete the rows with high missing value percentage

Before deleting the rows, we first observe the missing value distribution for each row. We can find that many rows have the missing value percentage greater than 30% for both label 0 and label 1.

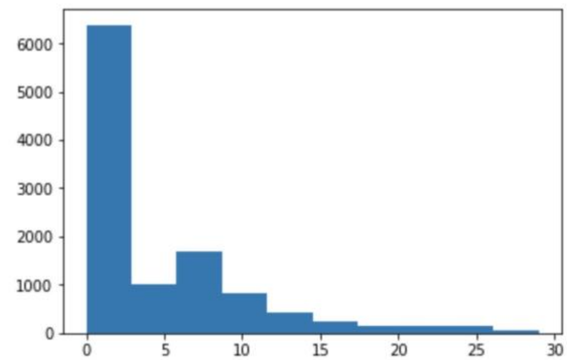


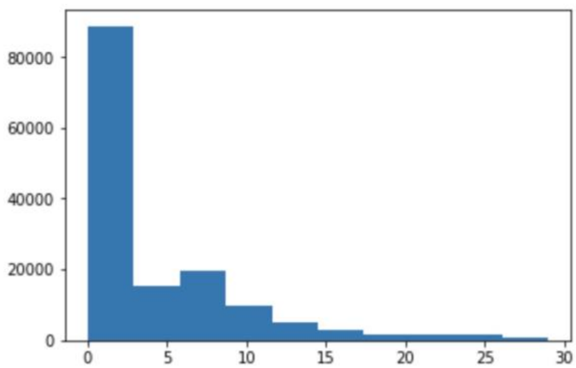
Missing data percentage for label zero



Missing data percentage for label one

So I decide to remove the rows who have the missing value percentage greater than 30%.





Missing data percentage for label zero

Missing data percentage for label one

Then after deleting the columns and rows with high missing data percentage, the distribution will look better than previous. We can see that the missing value percentage decreases in the whole dataset. We now have a 146794*105 dataset.

	Missing value counts	Missing value percentage
LANDAREA_AVG	182590	0.593767
LANDAREA_MODE	182590	0.593767
LANDAREA_MEDI	182590	0.593767
BASEMENTAREA_AVG	179943	0.585160
BASEMENTAREA_MODE	179943	0.585160
BASEMENTAREA_MEDI	179943	0.585160
EXT_SOURCE_1	173378	0.563811
NONLIVINGAREA_AVG	169682	0.551792
NONLIVINGAREA_MEDI	169682	0.551792
NONLIVINGAREA_MODE	169682	0.551792
ELEVATORS_MEDI	163891	0.532960
ELEVATORS_AVG	163891	0.532960
ELEVATORS_MODE	163891	0.532960
WALLSMATERIAL_MODE	156341	0.508408
APARTMENTS_MODE	156061	0.507497

Missing data percentage before data cleaning

	Missing value counts	Missing value percentage
EXT_SOURCE_1	86147	0.543648
OCCUPATION_TYPE	49616	0.313112
LANDAREA_AVG	33541	0.211667
LANDAREA_MODE	33541	0.211667
LANDAREA_MEDI	33541	0.211667
BASEMENTAREA_AVG	30896	0.194975
BASEMENTAREA_MODE	30896	0.194975
BASEMENTAREA_MEDI	30896	0.194975
EXT_SOURCE_3	29858	0.188425
NONLIVINGAREA_AVG	20633	0.130209
NONLIVINGAREA_MEDI	20633	0.130209
NONLIVINGAREA_MODE	20633	0.130209
AMT_REQ_CREDIT_BUREAU_YEAR	19548	0.123362
AMT_REQ_CREDIT_BUREAU_HOUR	19548	0.123362
AMT_REQ_CREDIT_BUREAU_DAY	19548	0.123362
AMT_REQ_CREDIT_BUREAU_WEEK	19548	0.123362

Missing data percentage after data cleaning

3)Data encoding

Data encoding is a method to transform the string value into numeric so that model is able to be implemented.

One hot encoding: it is used when the number of values in a feature is greater than 2. There are 11 features we need to use one hot encoding: 'CODE_GENDER', 'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE', 'WEEKDAY_APPR_PROCESS_START', 'ORGANIZATION_TYPE', 'HOUSETYPE_MODE', 'WALLSMATERIAL_MODE

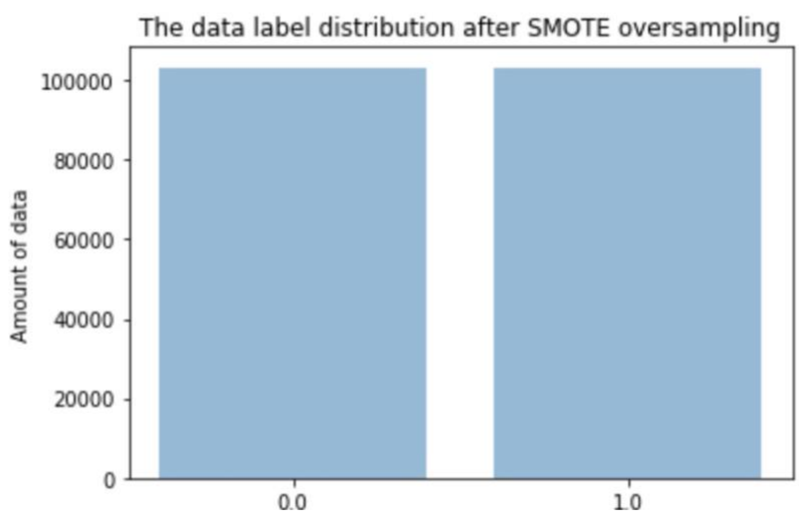
Data encoding: it is used when the number of values in a feature is less than 3. There are 4 features that need data encoding: NAME_CONTRACT_TYPE, FLAG_OWN_CAR, FLAG_OWN_REALTY, EMERGENCYSTATE_MODE

4)Impute data

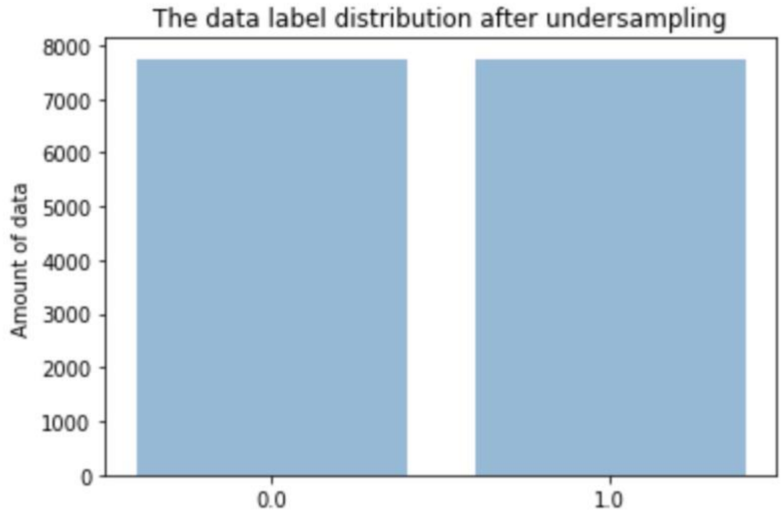
I impute the data by median value. Since I find that most of features have a very variance among their data.

5) Over sampling & Under sampling

I tried two different methods: Smote oversampling and under sampling. For the smote oversampling, it oversamples the minority data to match the majority data. For undersampling, it undersamples the majority data to match the minority data.



The data distribution after smote oversampling



The data distribution after Undersampling

6)Feature Selection

First, I removed the collinear features with the correlation greater than 0.9. The features we drop ped are:
'AMT_GOODS_PRICE',
'FLAG_EMP_PHONE',
'REGION_RATING_CLIENT_W_CITY'
'APARTMENTS_MODE', 'BASEMENTAREA_MODE'
'YEARS_BEGINEXPLUATATION_MODE'
'ELEVATORS_MODE', 'ENTRANCES_MODE'
'FLOORSMAX_MODE,
'LANDAREA_MODE', 'LIVINGAREA_MODE'
'NONLIVINGAREA_MODE'
'APARTMENTS_MEDI'
'BASEMENTAREA_MEDI'
YEARS_BEGINEXPLUATATION_MEDI'
'ELEVATORS_MEDI'
'ENTRANCES_MEDI'
'FLOORSMAX_MEDI'
LANDAREA_MEDI'
'LIVINGAREA_MEDI'
'NONLIVINGAREA_MEDI'
'TOTALAREA_MODE'
'OBS_60_CNT_SOCIAL_CIRCLE'
'CODE_GENDER_M'
'NAME_INCOME_TYPE_Pensioner'
'ORGANIZATION_TYPE_XNA'
(26 features)

The features we reserved are:

['AMT_CREDIT'],
['DAYS_EMPLOYED'],
['REGION_RATING_CLIENT'],
['APARTMENTS_AVG'],
['BASEMENTAREA_AVG']
['YEARS_BEGINEXPLUATATION_AVG']
['ELEVATORS_AVG']
[['ENTRANCES_AVG'
['FLOORSMAX_AVG']
['LANDAREA_AVG']
['LIVINGAREA_AVG']
['NONLIVINGAREA_AVG']
['APARTMENTS_AVG', 'APARTMENTS_MODE'],
['BASEMENTAREA_AVG', 'BASEMENTAREA_MODE'],
['YEARS_BEGINEXPLUATATION_AVG', 'YEARS_BEGINEXPLUATATION_MODE'],
['ELEVATORS_AVG', 'ELEVATORS_MODE'],
['ENTRANCES_AVG', 'ENTRANCES_MODE'],
['FLOORSMAX_AVG', 'FLOORSMAX_MODE'],
['LANDAREA_AVG', 'LANDAREA_MODE'],
['LIVINGAREA_AVG', 'LIVINGAREA_MODE'],
['NONLIVINGAREA_AVG', 'NONLIVINGAREA_MODE'],
['LIVINGAREA_AVG', 'LIVINGAREA_MODE', 'LIVINGAREA_MEDI']
['OBS_30_CNT_SOCIAL_CIRCLE']
['CODE_GENDER_F']
['DAYS_EMPLOYED', 'FLAG_EMP_PHONE']
['DAYS_EMPLOYED', 'FLAG_EMP_PHONE', 'NAME_INCOME_TYPE_Pensioner']

7)Remove no-importance features using lightgbm

I used lightgbm in the exploratory data analysis and removed the features with no importance in this model. There are 105 features I remove using lightgbm. Most of the features are from the one-hot-encoding

8)Add features using domain knowledge

CREDIT_INCOME_PERCENT = AMT_CREDIT / AMT_INCOME_TOTAL

ANNUITY_INCOME_PERCENT = AMT_ANNUITY / AMT_INCOME_TOTAL

DAYS_EMPLOYED_PERCENT = DAYS_EMPLOYED / DAYS_BIRTH

CREDIT_TERM = AMT_ANNUITY / AMT_CREDIT

3. Modeling

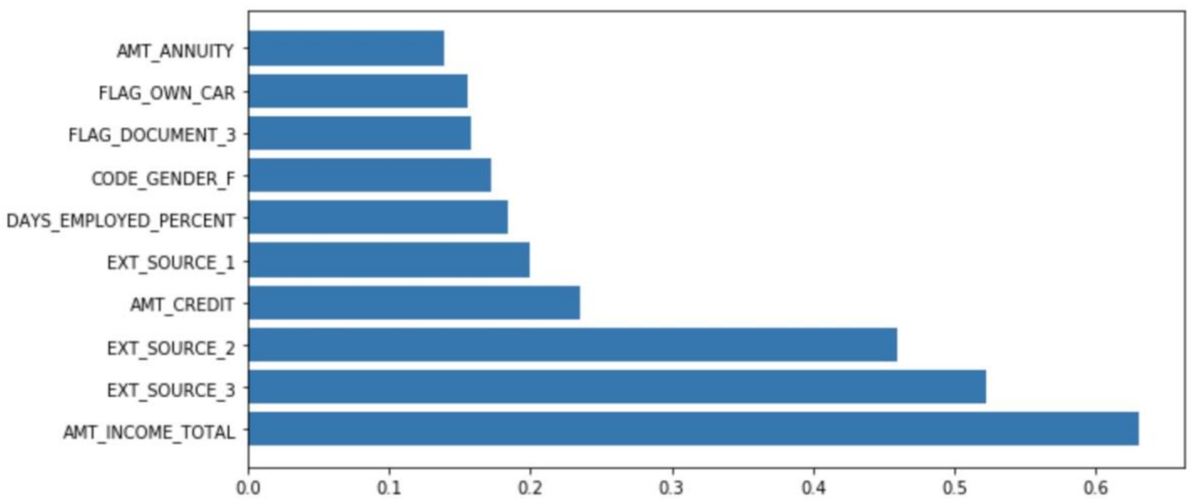
After preprocessing, we get three different datasets for modeling: Undersampled dataset, Smote oversampled dataset, and not sampled dataset.

We use six measurement for the performance our model: Testing accuracy, Cross Validation, Recall, Precision, and F1 score. And among these six measurements, we will focus on more on recall, precison, and F1 score. Recall means that how many people with bad loan are selected, precision means that how relevant is the data selected. And F1 score is the combination of recall and precision. Since we want to find the bad loan person and don't want to refuse people who can pay the loan on time, we would focus on F1 score most.

We will implement five different models: SVM, KNN, Logistic Regression, Random Forest, Neural Network.

1) **Support Vector Machine**

- **Feature importance**



- **Original model performance**

	Name	Test Accuracy	Cross-Validation Accuracy
0	Dataset using smote oversampling	0.6178	0.874734
1	Dataset without oversampling	0.9265	0.929143
2	Dataset with undersampling	0.5676	0.668129

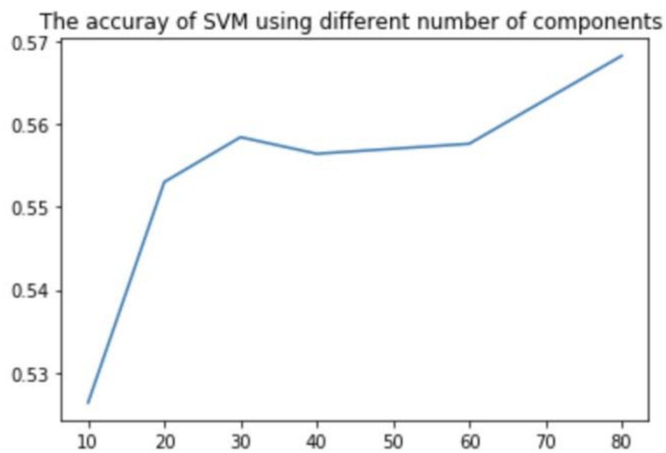
	Precision	F1 score	Recall
0	0.112432	0.191966	0.656069
1	1.000000	0.004515	0.002262
2	0.124791	0.216667	0.821429

For dataset without oversampling, the accuracy is the highest since the number of label 1 and label 0 is 1:10. We can find that using dataset with undersampling has a high recall, which means it can select 82% bad loan person from the entire people. But its precision is very low, it means it mistakenly select many people who don't have bad loan.

- **Feature selection using PCA**

The performace of SVM with PCA feature selection is:

	PCs	Test Accuracy	Cross-Validation Accuracy	Precision	F1 score	Recall
0	10	0.5264	0.616968	0.103639	0.181189	0.719780
1	20	0.5530	0.629548	0.110371	0.191682	0.728022
2	30	0.5584	0.638323	0.115192	0.200000	0.758242
3	40	0.5564	0.642000	0.119458	0.207857	0.799451
4	60	0.5576	0.655355	0.120066	0.208870	0.802198
5	80	0.5682	0.666581	0.124005	0.215194	0.813187



From the graph, we can see that as the number of principle component increases, recall increases, F1 score increases

- **Feature selection using feature importance**

	Feature num	Test Accuracy	Cross-Validation Accuracy	Precision	F1 score	Recall
0	10	0.5662	0.666581	0.118737	0.206367	0.787709
1	20	0.5668	0.668000	0.119529	0.207754	0.793296
2	30	0.5626	0.668710	0.121951	0.212459	0.824022
3	40	0.5548	0.667806	0.117840	0.205567	0.804469
4	60	0.5522	0.666258	0.114391	0.199499	0.779330

When we use feature importance to do feature selection, we can find the precison, recall ,and f1 score are the highest when we select the first 30 features

2) KNN

- **Original model performance**

	Name	Test Accuracy	Cross-Validation Accuracy	\
0	Dataset using smote oversampling	0.3638	0.693933	
1	Dataset without oversampling	0.9220	0.930286	
2	Dataset with undersampling	0.6258	0.583548	

	Precision	F1 score	Recall
0	0.086166	0.156010	0.823529
1	0.000000	NaN	0.000000
2	0.104123	0.173951	0.528150

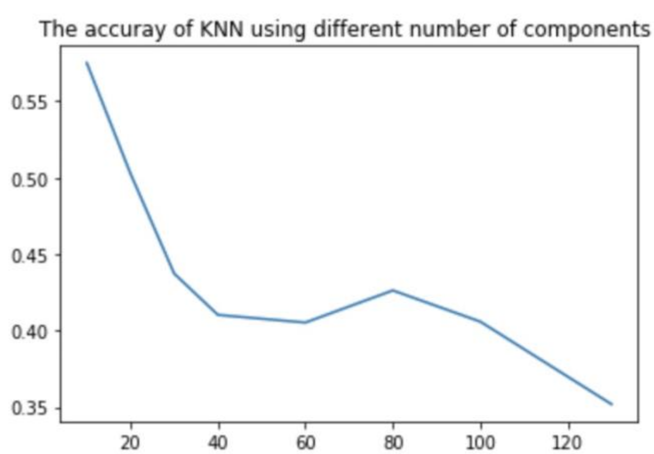
We can see that using dataset with smote oversampling will have a high recall, but a lower F1 score.

- **Feature selection using PCA**

The performace of KNN with PCA feature selection is:

	PCs	Test Accuracy	Cross-Validation Accuracy	Precision	F1 score	\
0	10	0.5750	0.674998	0.097693	0.168948	
1	20	0.5026	0.699197	0.094661	0.167392	
2	30	0.4372	0.693133	0.090033	0.161502	
3	40	0.4102	0.693332	0.087219	0.157188	
4	60	0.4052	0.700067	0.083914	0.151256	
5	80	0.4262	0.709337	0.085442	0.153438	
6	100	0.4058	0.706470	0.081607	0.147000	
7	130	0.3518	0.691270	0.078111	0.141912	

	Recall
0	0.624277
1	0.722543
2	0.783237
3	0.794798
4	0.765896
5	0.751445
6	0.739884
7	0.774566



We can see that the when we use the first 10 principle components, we will have the highest F1 score, whereas our recall is lower.

3) Logistic Regression

- **Original model performance**

	Name	Test Accuracy	Cross-Validation Accuracy	Precision	F1 score	Recall
0	Dataset using smote oversampling	0.562400	0.697733	0.118383	0.206096	0.795518
1	Dataset without oversampling	0.920667	0.929928	0.318182	0.028571	0.014957
2	Dataset with undersampling	0.569000	0.672065	0.120528	0.208012	0.758713

When we logistic regression, we can see that using dataset with smote oversampling will give us a better F1 score and recall

- **Feature selection using PCA**

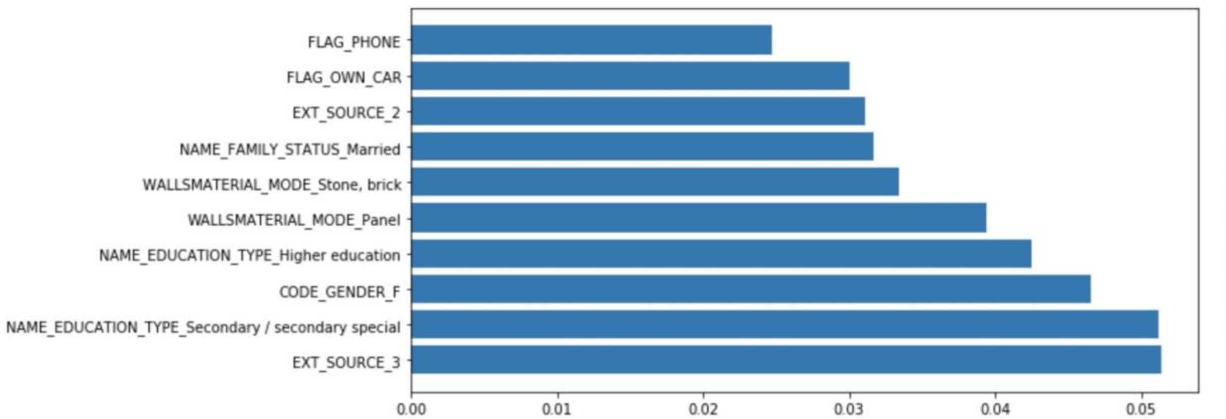
The performace of KNN with PCA feature selection is:

	PCs	Test Accuracy	Cross-Validation Accuracy	Precision	F1 score	Recall
0	10	0.5294	0.629333	0.104142	0.183270	0.763006
1	20	0.5464	0.646268	0.106792	0.187097	0.754335
2	30	0.5474	0.656536	0.106366	0.186264	0.748555
3	40	0.5516	0.660334	0.107291	0.187681	0.748555
4	60	0.5646	0.675135	0.114201	0.199338	0.783237
5	80	0.5664	0.682002	0.115612	0.201767	0.791908
6	100	0.5644	0.692269	0.115772	0.202198	0.797688
7	130	0.5652	0.699466	0.116611	0.203663	0.803468

When using the 130 principle components, we have the highest F1 score and recall

4) Random Forest

- **Feature importance**



We can first observe the feature importance of random forest. The ext_source_3 is the most important feature

- **Original model performance**

	Name	Test Accuracy	Cross-Validation Accuracy
0	Dataset using smote oversampling	0.562400	0.697733
1	Dataset without oversampling	0.927333	0.935572
2	Dataset with undersampling	0.685400	0.672323

	Precision	F1 score	Recall
0	0.118383	0.206096	0.795518
1	1.000000	0.128000	0.068376
2	0.139856	0.228543	0.624665

When using the dataset with undersampling, we have the highest precison and F1 score

5) Neural Network

- **Original model performance**

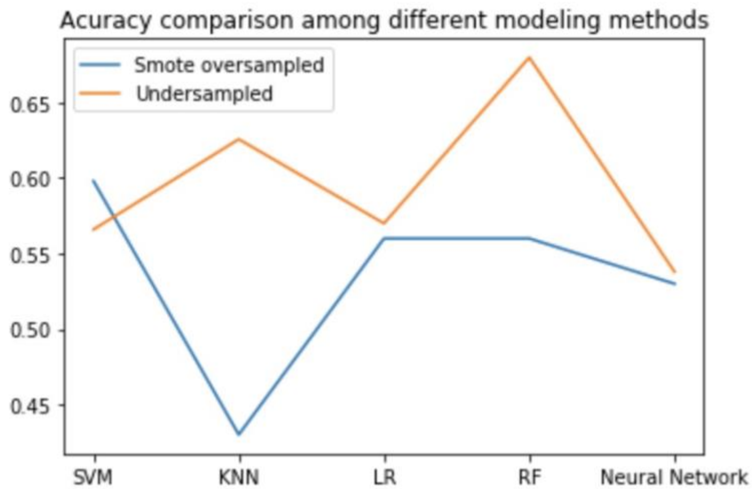
	Name	Test Accuracy	Cross-Validation Accuracy
0	Dataset using smote oversampling	0.532800	0.766000
1	Dataset without oversampling	0.891667	0.899859
2	Dataset with undersampling	0.538200	0.632194

	Precision	F1 score	Recall
0	0.088843	0.155459	0.621387
1	0.182927	0.155844	0.135747
2	0.110220	0.192375	0.755495

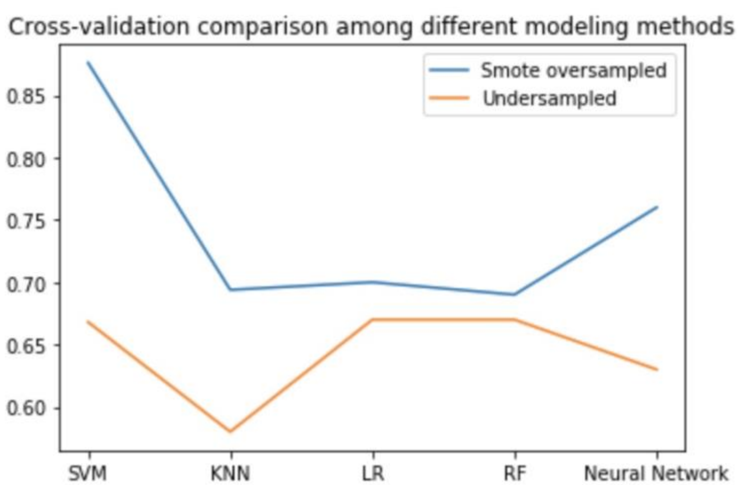
We can see that when using neural network, it does not work as well as other models

6) Model Comparison

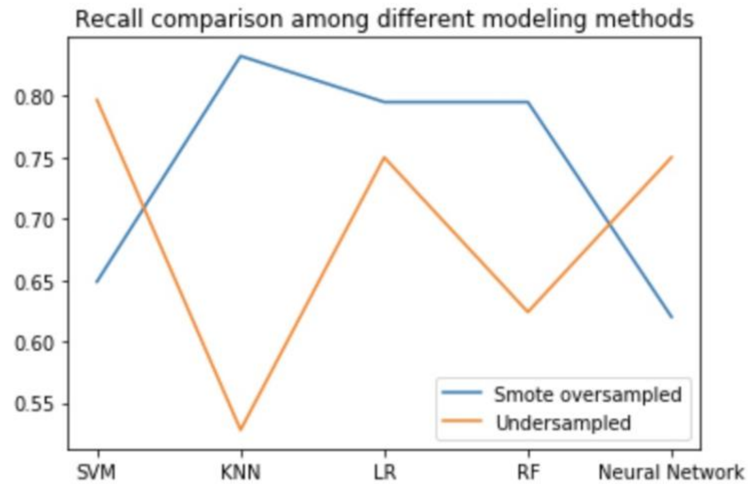
Testing accuracy



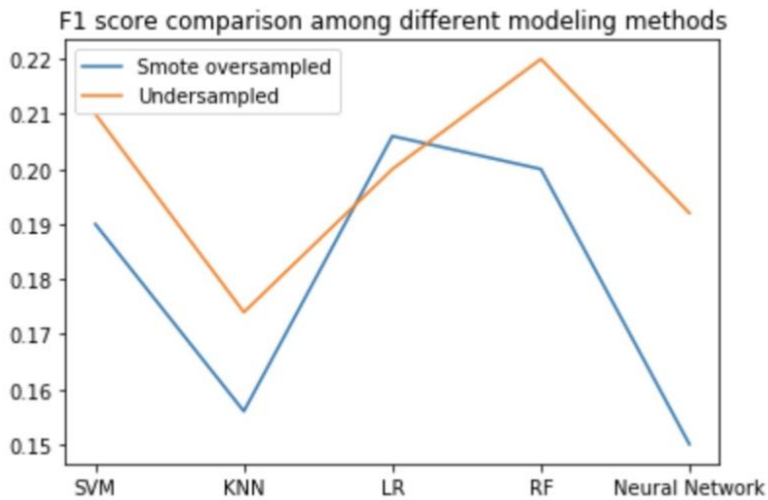
Cross Validation



Recall



F1 score



From the graphs, we can see that KNN and RF have the best testing accuracy and recall. But the F1 score of KNN is low. In order to keep a balance between recall and precision, we choose to select the model with the highest F1 score: SVM, Random Forest, Logistic Regression

4. Model tuning

1) SVM

For SVM, we tune the parameters of C value and gamma value

	Gamma value	Accuracy	Recall	Precision	F1
0	0.001	0.673290	0.658065	0.678765	0.673199
1	0.010	0.668516	0.665548	0.669489	0.668476
2	0.100	0.578129	0.797806	0.554402	0.556561
3	1.000	0.500645	0.898323	0.480453	0.337371

	C value	Accuracy	Recall	Precision	F1
0	0.001	0.642710	0.752258	0.617177	0.638278
1	0.010	0.642710	0.752258	0.617177	0.638278
2	0.100	0.667226	0.646065	0.674550	0.667037
3	1.000	0.673290	0.658065	0.678765	0.673199
4	10.000	0.674774	0.665290	0.678232	0.674708

Since we focus most on F1 score, we set gamma=0.001 and C=10 to have the best F1 score 0.674

2) Random Forest

We tune the number of trees and max depth for the random forest

	Num_trees	Accuracy	Recall	Precision	F1
0	50	0.662387	0.635226	0.669515	0.662000
1	100	0.672129	0.653290	0.678157	0.673409
2	500	0.678000	0.669161	0.681245	0.676639
3	1000	0.677806	0.671742	0.681831	0.677671

	Max_depth	Accuracy	Recall	Precision	F1
0	5	0.667548	0.671355	0.669030	0.669846
1	10	0.676516	0.679355	0.676044	0.679002
2	20	0.680323	0.674968	0.679288	0.679244
3	30	0.677871	0.672000	0.680422	0.679233

When num trees=1000, max depth=20, we have the highest F1 score 0.6792

3) Logistic Regression

We tune the C value for LR

	C value	Accuracy	Recall	Precision	F1
0	0.01	0.671806	0.660129	0.676007	0.671743
1	0.10	0.671871	0.661290	0.675687	0.671815
2	1.00	0.672065	0.662065	0.675647	0.672012
3	10.00	0.671935	0.661935	0.675510	0.671884

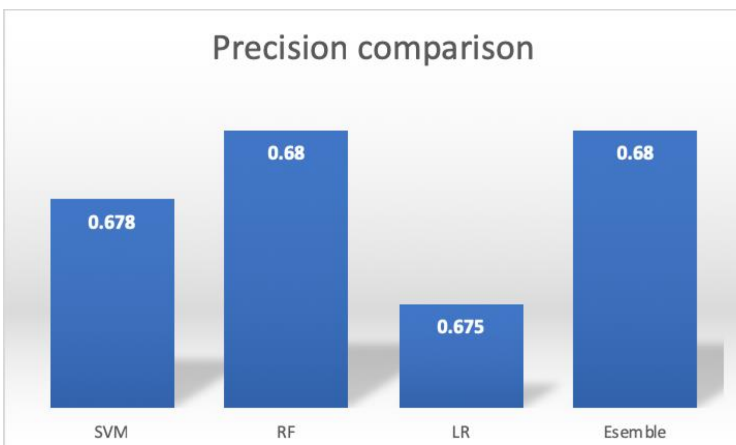
When c value=1, we have the highest F1 socre 0.672

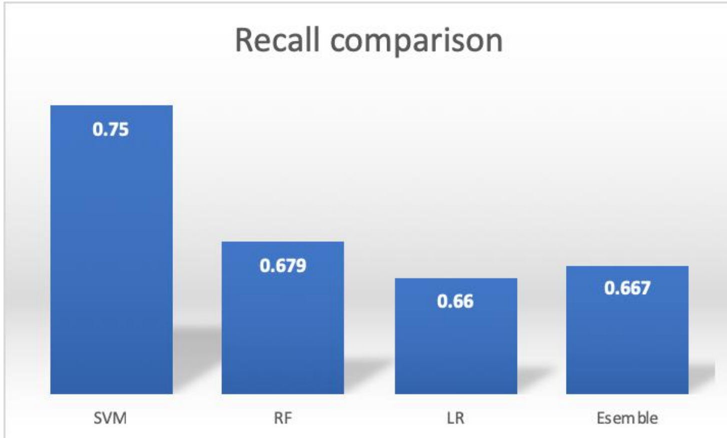
5. Ensembling

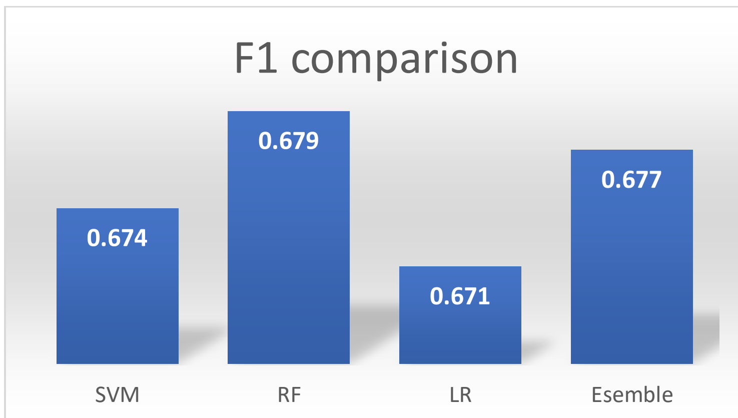
We use majority vote as our ensembling method. And we use LR,SVM and RF as our models

```
Accuracy 0.6763870967741935
Recall 0.6668387096774193
Precision 0.6804633858046878
F1 0.6770432334755305
```

Performance comparison:







We can see that SVM has a really good performance in recall, which means it does well in selecting the bad loan people from the entire people. But Random Forest and Esemble have the best precison, which means the person they selecte are more likely to be a bad loan person. Random Forest and Esemble also have the best F1 score.

6. Conclusion

1) When we want to predict minority data, we will focus on more on the precision, recall and F1 score instead of the accuracy

2) If we want to recognize most person who have repay problems, we should SVM

3) If we want to recognize the person who have repay problems as accurate as possible, we should use Random Forest

4) If we want to consider both situations, we also would consider Random Forest