CREDIT DEFAULT PREDICTION

Amey Mhadgut | Luke | Sky |Lidia

HOME CREDIT GROUP

Business Understanding

- Home Credit is an international **lender for home mortgages** with operations in 9 countries
- Focuses on responsible lending primarily to **unbanked** population
- Predicting a client's **repayment ability** is a critical business need
- Historical data of **~300k** manually processed loan applications
- Business Action: Decide **whether to fund a loan** based on **probability of default**

Predict **probability of default** to support the decision process

OUR GOAL

Business Action: Will we fund a given applicant

Assumptions:

1. Same premium interest rate to all applicants
2. Principle determined by applicant
3. Loan term is 30 years

DATA IN HAND

Highly unbalanced, missing values, unclean data

~300k
Historical loan applications

9.6%
Low base rate

121
Features

~200
Dimensionality

Target Variable
1 - Applicant defaulted on the loan
0 - Applicant didn't default on the loan

Data Types
- Nominal
- Numeric

Data File Format
.csv

Some features

Feature Name	Description
AMT_INCOME_TOTAL	Income of the client
DAYS_EMPLOYED	How many days before the application the person started current employment
FLAG_OWN_REALTY	Flag if client owns a house or flat
DAYS_BIRTH	Client's age in days at the time of application
CNT_FAM_MEMBERS	How many family members does client have
EXT_SOURCE_1 EXT_SOURCE_2 EXT_SOURCE_3	Normalized score from external data source

Data Cleaning

Data cleaning & preparation

Raw data .csv

- Improper datatypes
- Raw format

→

Data cleaning

- Resampling without replacement
- Datatype correction
- CSV to Weka format

→

Data ready for Weka

- .arff
- Correct datatypes
- ~50k instances

* Refer Appendix A for more details

DATA MODELS

Test Method: **Spilt by % 66/33**

Classifier	Accuracy (%)	AUC	Confusion Matrix
Logistic Regression*	**91.67**	**0.732**	a b <-- classified as 15557 33 | a = 0 1383 17 | b = 1
Support Vector Machine	91.76	0.500	a b <-- classified as 15590 0 | a = 0 1400 0 | b = 1
Random Forest	91.75	0.700	a b <-- classified as 15590 0 | a = 0 1400 0 | b = 1
Decision Tree (J48)	91.76	0.500	a b <-- classified as 15590 0 | a = 0 1400 0 | b = 1
Naïve Bayes	72.56	0.665	a b <-- classified as 11666 3924 | a = 0 737 663 | b = 1

* Refer Appendix B for more details

FINAL MODEL

Logistic Regression

Why Logistic Regression?

Considering our use case, we need **well calibrated probabilities** & logistic regression returns that.

Other benefits:

- Higher **Area Under ROC**
- Easy to iteratively **train** with new data
- Relatively **faster** when applying

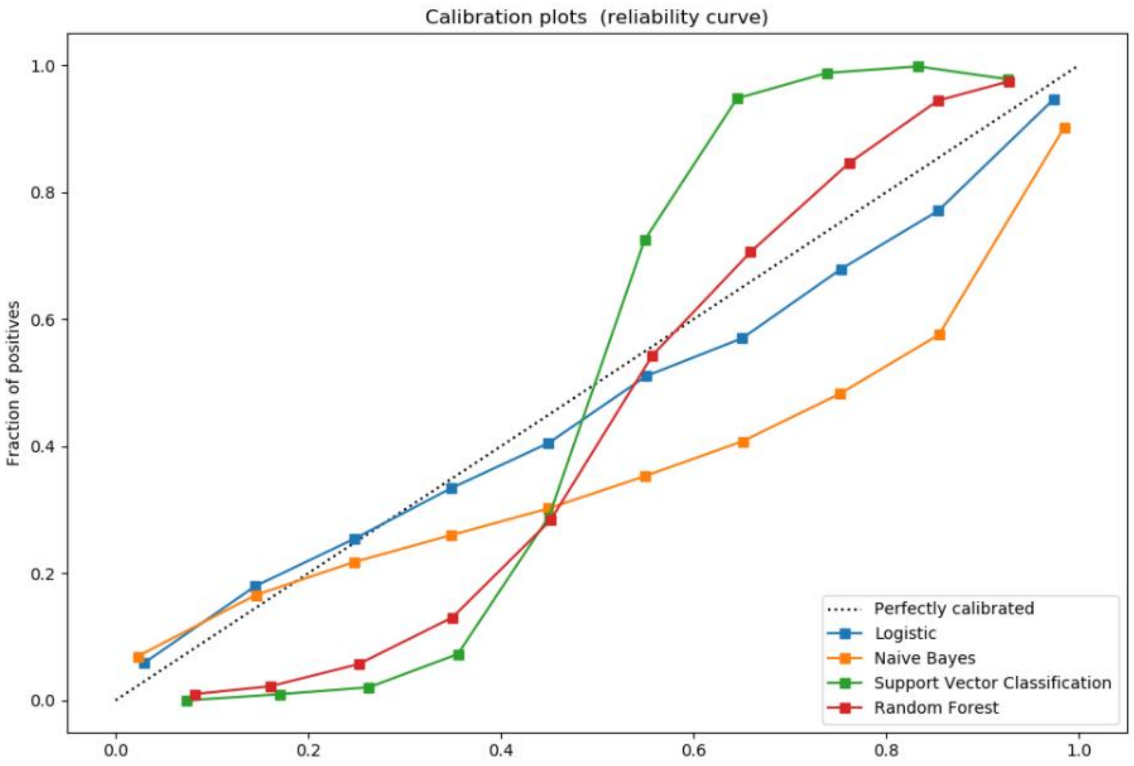


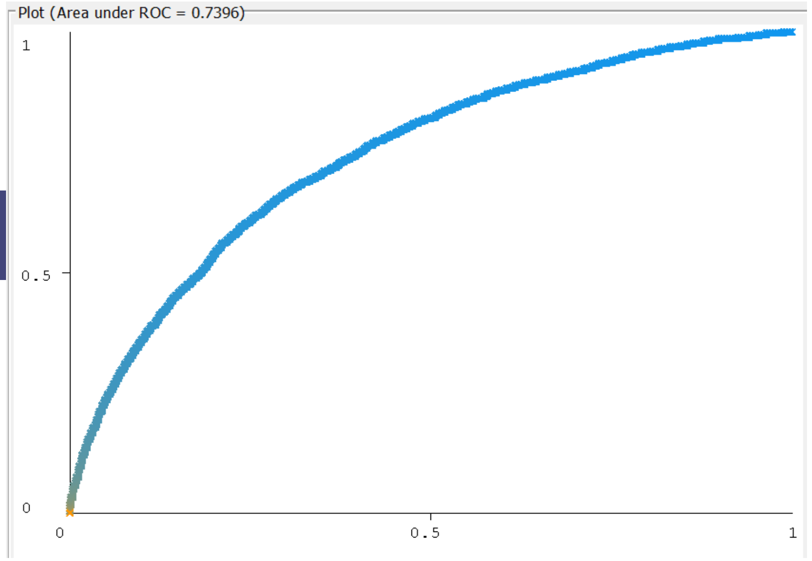
Fig. Comparison of Probability calibration plot for different classifiers

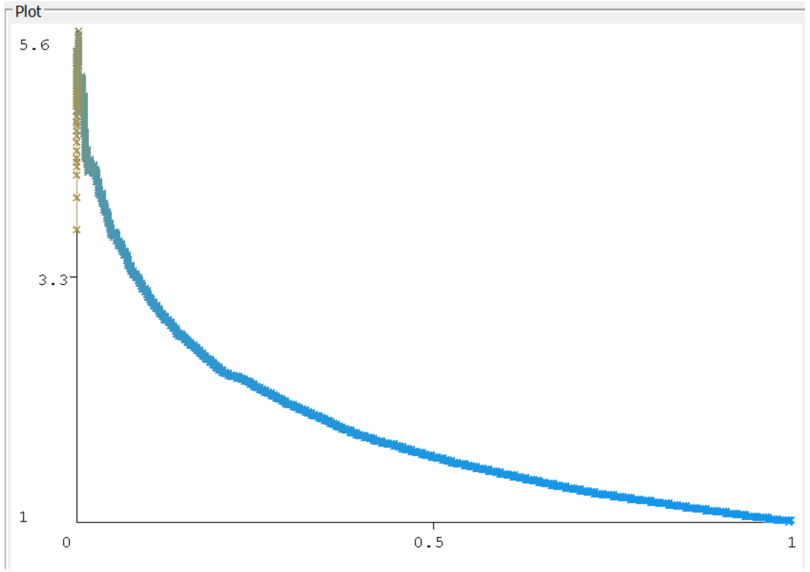
FINAL MODEL

Logistic Regression – CV (10-folds)

```
Correctly Classified Instances       45830               91.715  %
Incorrectly Classified Instances      4140                8.285  %
Kappa statistic                          0.0223
Mean absolute error                      0.1397
Root mean squared error                  0.2655
Relative absolute error                 92.2656 %
Root relative squared error             96.4686 %
Total Number of Instances            49970

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.986    0.918      0.998    0.957      0.067    0.740     0.964     0
                 0.014    0.002    0.439      0.014    0.027      0.067    0.740     0.218     1
Weighted Avg.    0.917    0.905    0.879      0.917    0.880      0.067    0.740     0.903

=== Confusion Matrix ===

     a     b   <-- classified as
 45772    74 |     a = 0
  4066    58 |     b = 1
```

Ridge	Generalization Accuracy (%)	AUC
1.00E+16	91.747	0.5
1.00E+8	91.747	0.601
1	91.659	0.732
1.00E-8	91.715	**0.74**
1.00E-16	91.747	0.601





Cut Off Determination

Best Cut Off for Max Profit

Assumptions:

- All loan amounts are 100k
- Annual Rate is 2.5%
- Term 30 years
- Default is at year 15

Change This...

To Maximize This...

actual	predicted	error	predictior	p(Non-Default)	Decision	Cash Flow, model	Cash Flow, Null	error	tp	fp	fn		CUTOFF	0.96			Rate	2.50%
1	1		0.959	0.959	1	1678.98	1678.98	0	1	0	0		NPV Model	$ 10,816,076.42			Principle	$ 100,000.00
1	1		0.985	0.985	1	1678.98	1678.98	0	1	0	0		NPV Null	$ (83,264,821.30)			Term (Mo)	360
1	1		0.994	0.994	1	1678.98	1678.98	0	1	0	0			-113%			Monthly Payment	$395.12
1	1		0.964	0.964	1	1678.98	1678.98	0	1	0	0						Annualized	$4,741.45
1	1		0.888	0.888	2	0.00	1678.98	1	0	0	1		% Funded	40%			NPV Non Default	$1,678.98
1	1		0.938	0.938	2	0.00	1678.98	1	0	0	1						NPV Default	($38,855.28)
1	1		0.898	0.898	2	0.00	1678.98	1	0	0	1							
1	1		0.976	0.976	1	1678.98	1678.98	0	1	0	0		Non-Default	Default	<- Predicted As			
1	1		0.867	0.867	2	0.00	1678.98	1	0	0	1		19610	26236	Non-Default			
1	1		0.847	0.847	2	0.00	1678.98	1	0	0	1		569	3555	Default			

DEPLOYMENT

Make data driven decisions

- Deploy a **Logistic Regression** model which will run on new applications
- Predict **Probability of default**
- Home credit group's loan department to decide on whether to fund based on **implied interest rate, probability of default, and cut off of .96***
- Apply the decision across the list of applications

* Refer Appendix C for more details

Deployment

Use Case Example

Cut Off = .96

Principle (From Applicant)

Interest Rate(From Lender)

MODEL
(Probability Est)

Fund? (Y/N)

Other Business Use:

- Modify premium rate based on market conditions
- Create a tiered interest rate system
- Cut Off for different term lengths (15 years)
- Building block for more complex business questions
  - Input for regression models -> EV of Cashflow
  - Optimize loan portfolio for reward/risk

Challenges:

- Potential for multiple input variables
- Timing Issue - When do they default? What was their original term?
- Non-Uniform Cash flow

KEY LEARNINGS

Data preparation can be tedious

Sometimes data can be in **huge volumes** and with a lot of garbage. Allot enough **time** for data preparation. Lot of data isn't always good in which case **sampling** can help.

Always keep in mind the use case

We found a strong tendency to drift between answering **regression** and **categorical** scenarios. Re-center around the use case to avoid going on analytical tangents.

Consider human bias

Ensure that you consider **human biases** while doing data science projects. In our case, the historical loan applications were approved by manual procedure and may involve human bias.

THANK YOU

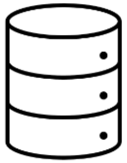
APPENDIX A

Feature Engineering – Lessons Learnt

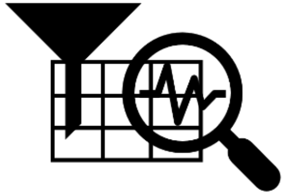
- Initially, we proposed feature selection over reducing the size of dataset
- However, we learnt that there are high chances of missing out on important features
- So, we picked 50k instances maintaining the same base rate
- Find previous feature selection approach in the following slides

APPENDIX A (CONTD.)

Proposed Feature selection, data cleaning & preparation



Raw data .csv
- Improper datatypes
- Multiple features
- Missing data



Feature selection & data cleaning*
- % Missing values
- Correlation w.r.t target
- Domain Knowledge
- Resampling (for SVM & Random Forest)

Data ready for Weka
- .arff
- Limited features
- Correct datatypes

* Refer Appendix A for more details

APPENDIX A (CONTD.)

Proposed Feature Engineering Process

Found top **features** based on:

- Correlation with respect to the target
- Domain knowledge
- Missing Values %

Action:

- Enlisted features with high missing value %
- Skimmed through to find any features of domain importance & high correlation to target, ignore these features
- Remove the rest

APPENDIX A (CONTD.)

Finding Missing Value %

Used the below code

```
#Importing Pandas & NumPy libs
import pandas as pd
import numpy as np

#Reading csv file
df = pd.read_csv(r"*\application_train.csv")

#Getting dataframe of attributes, it's sum of missing values
a=pd.DataFrame(df.isnull().sum(),
columns=["missing_values"])

#Calculating & adding % of missing values
a["% of missing values"]=
np.round((df.isnull().sum()*100)/len(df),2)

#Sort asc
b=a.sort_values("missing_values", ascending=False)

#Display all rows without truncation
pd.set_option('display.max_rows', None)
#Print the results
print(b)
```

The number of columns with missing values are 67 out of 122 columns

	missing_values	% of missing values
COMMONAREA_MEDI	214865	69.87
COMMONAREA_AVG	214865	69.87
COMMONAREA_MODE	214865	69.87
NONLIVINGAPARTMENTS_MODE	213514	69.43
NONLIVINGAPARTMENTS_AVG	213514	69.43
NONLIVINGAPARTMENTS_MEDI	213514	69.43
FONDKAPREMONT_MODE	210295	68.39
LIVINGAPARTMENTS_MODE	210199	68.35
LIVINGAPARTMENTS_AVG	210199	68.35
LIVINGAPARTMENTS_MEDI	210199	68.35
FLOORSMIN_AVG	208642	67.85
FLOORSMIN_MODE	208642	67.85
FLOORSMIN_MEDI	208642	67.85
YEARS_BUILD_MEDI	204488	66.50
YEARS_BUILD_MODE	204488	66.50
YEARS_BUILD_AVG	204488	66.50
OWN_CAR_AGE	202929	65.99
LANDAREA_MEDI	182590	59.38
LANDAREA_MODE	182590	59.38
LANDAREA_AVG	182590	59.38
BASEMENTAREA_MEDI	179943	58.52
BASEMENTAREA_AVG	179943	58.52
BASEMENTAREA_MODE	179943	58.52
EXT_SOURCE_1	173378	56.38
NONLIVINGAREA_MODE	169682	55.18
NONLIVINGAREA_AVG	169682	55.18
NONLIVINGAREA_MEDI	169682	55.18
ELEVATORS_MEDI	163891	53.30
ELEVATORS_AVG	163891	53.30
ELEVATORS_MODE	163891	53.30
WALLSMATERIAL_MODE	156341	50.84
APARTMENTS_MEDI	156061	50.75
APARTMENTS_AVG	156061	50.75
APARTMENTS_MODE	156061	50.75
ENTRANCES_MEDI	154828	50.35
ENTRANCES_AVG	154828	50.35
ENTRANCES_MODE	154828	50.35
LIVINGAREA_AVG	154350	50.19
LIVINGAREA_MODE	154350	50.19
LIVINGAREA_MEDI	154350	50.19
HOUSETYPE_MODE	154297	50.18
FLOORSMAX_MODE	153020	49.76
FLOORSMAX_MEDI	153020	49.76
FLOORSMAX_AVG	153020	49.76
YEARS_BEGINEXPLUATATION_MODE	150007	48.78
YEARS_BEGINEXPLUATATION_MEDI	150007	48.78
YEARS_BEGINEXPLUATATION_AVG	150007	48.78
TOTALAREA_MODE	148431	48.27
EMERGENCYSTATE_MODE	145755	47.40
OCCUPATION_TYPE	96391	31.35
EXT_SOURCE_3	60965	19.83

APPENDIX A (CONTD.)

Finding correlation w.r.t. target variable

Used Weka

Step 1: Go to Select Attributes tab

Step 2: Select CorrelationAttributeEval

Step 3: Select Target variable

Step 4: Click Start

This will give a list of attributes with correlation.

```
=== Attribute Selection on all input data ===

Search Method:
    Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 2 TARGET):
    Correlation Ranking Filter
Ranked attributes:
 0.160303    41 EXT_SOURCE_2
 0.157397    42 EXT_SOURCE_3
 0.078239    18 DAYS_BIRTH
 0.055218    47 DAYS_LAST_PHONE_CHANGE
 0.054706     4 CODE_GENDER
 0.051457    21 DAYS_ID_PUBLISH
 0.050994    38 REG_CITY_NOT_WORK_CITY
 0.049404    14 NAME_EDUCATION_TYPE
 0.045982    23 FLAG_EMP_PHONE
 0.044932    19 DAYS_EMPLOYED
```

APPENDIX A (Contd.)

For non-linear models

For SVM & Random Forest, due to the **performance issues**, we used **resampling with replacement** technique to get a smaller sample dataset

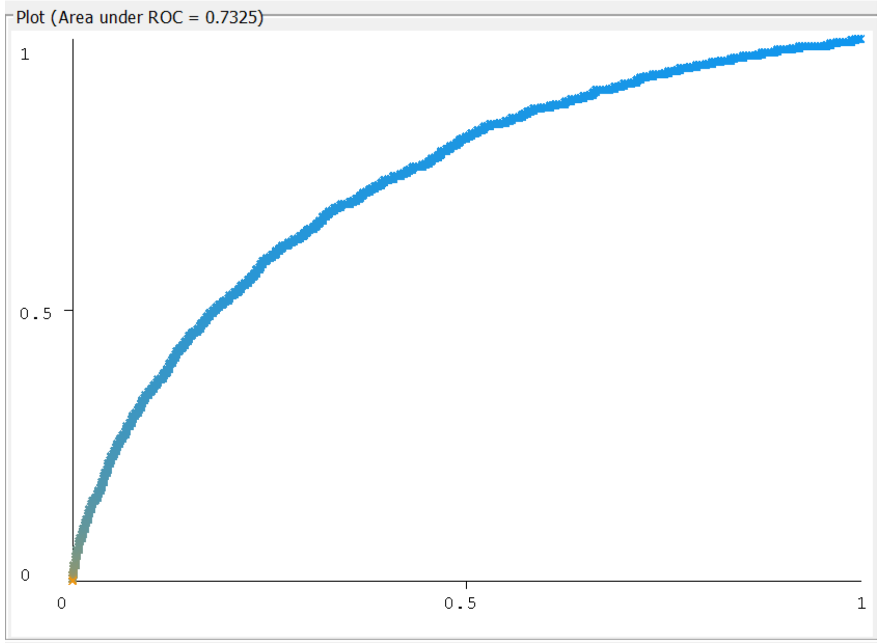
Steps:

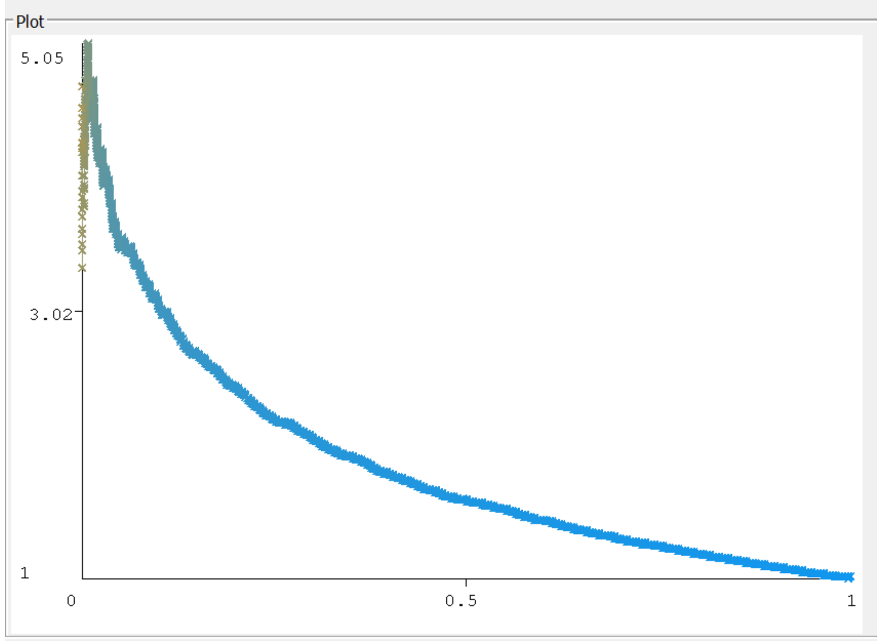
- In Weka, apply the Filter: Resample
- Set noReplacement = true and percentage
- Apply

APPENDIX B

Logistic Regression – Spilt by % (66/33)

```
=== Summary ===

Correctly Classified Instances       15574               91.6657 %
Incorrectly Classified Instances      1416                8.3343 %
Kappa statistic                          0.0179
Mean absolute error                      0.1398
Root mean squared error                  0.2661
Relative absolute error                 92.3428 %
Root relative squared error             96.7866 %
Total Number of Instances            16990

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.988    0.918      0.998    0.956      0.051    0.732     0.962     0
                 0.012    0.002    0.340      0.012    0.023      0.051    0.732     0.211     1
Weighted Avg.    0.917    0.907    0.871      0.917    0.880      0.051    0.732     0.900

=== Confusion Matrix ===

     a     b   <-- classified as
 15557    33 |     a = 0
  1383    17 |     b = 1
```





APPENDIX C

Explanation – Picking up the cut-off

- Export predictions into Excel
- Determine our model’s probability estimate for “Non-Default”
- Given assumptions of same loan offer to all applicants and defaults occurring at the same point in term of loan, calculate NPV of a Non Default and NPV of a Default
- Decision to fund changes based on the Cut Off established
- Run optimization to maximize the NPV for sum of all applicants by adjusting the Cut Off
- Value of NPV Model is not a prediction! This technique was used to account for difference between the **cost** of a default vs. the **benefit** of a non-default.