

SC1015 PRESENTATION

By: Jerald (U2221612D)

Bryan (U2222849J)

Jonathan (U2223191G)

TABLE OF CONTENTS



Motivations



MACHINE LEARNING

Decision Tree, Random Forest, K-Nearest Neighbours, Naive Bayes



Visualisation and Cleaning



Our Thoughts

CSV DATA SET

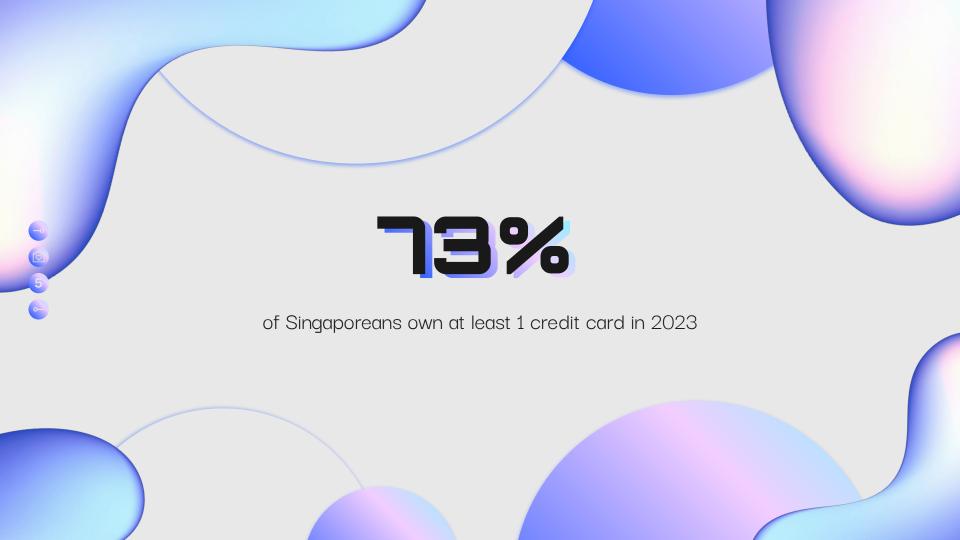
MOTIVATION

- Cashless payments are increasingly common with the push of such technology
- To **minimise** the **risk** of lending money to customers who may not pay back their loans.
- To understand which **factors** affect people from defaulting credit card loans.









DATA SET



application_data
columns_description
previous_application



19/124 Columns in application_data

TARGET | NAME_CONTRACT_TYPE |
FLAG_OWN_CAR | FLAG_OWN_REALTY |
AMT_CREDIT | AMT_GOODS_PRICE, etc.

PROBLEM DEFINITION

"To find the **highest correlation** of credit card defaulters and to **predict** loan default risk of defaulters."

DATA CLEANING

SPLITTING THE DATA



- Finding **NULL** values within dataset
- Fill in missing values with MEDIAN

[] refined_app.isnull().sum() SK_ID_CURR	0	AMT_ANNUITY 12 AMT_GOODS_PRICE 278 NAME_TYPE_SUITE 1292 NAME_INCOME_TYPE 0 DAYS BIRTH 0
TARGET NAME_CONTRACT_TYPE FLAG_OWN_CAR FLAG_OWN_REALTY CNT_CHILDREN AMT_INCOME_TOTAL AMT_CREDIT	0 0 0 0 0	DAYS_EMPLOYED 0 OCCUPATION_TYPE 96391 OBS_30_CNT_SOCIAL_CIRCLE 1021 DEF_30_CNT_SOCIAL_CIRCLE 1021 OBS_60_CNT_SOCIAL_CIRCLE 1021 DEF_60_CNT_SOCIAL_CIRCLE 1021 dtype: int64



- Finding **empty values** within dataset
- Fill in missing values with **NULL**

```
.isnull().sum()
```

FILLING MISSING VALUES

NUMERICAL

6/13 columns with missing data

```
def median_impute(df,col):
    return df[col].fillna(df[col].median())

refined_num['AMT_ANNUITY'] = median_impute(refined_num,'AMT_ANNUITY')
```



Filling in with **MEDIAN**

CATEGORICAL

2/6 columns with missing data

NAME_TYPE_SUITE: Fill with "Unaccompanied"

OCCUPATION_TYPE: Fill with "NILL"



REMOVING DUPLICATES

- Ensure each record represents a unique observation
- Affects data analysis:
 - Skewness of results of statistical analysis
 - Increasing size of data set unnecessarily
 - Overfitting in machine learning



```
[ ] df = refined_app.copy()
    df.drop_duplicates(inplace=True)
    print('The amount of frauds in df before dropping duplicates:', len(refined_app[refined_app['TARGET'] == 1]))
    print('The amount of frauds in df after dropping duplicates:', len(df[df['TARGET'] == 1]))

The amount of frauds in df before dropping duplicates: 24825
The amount of frauds in df after dropping duplicates: 24825
```



NUMERICAL DATA

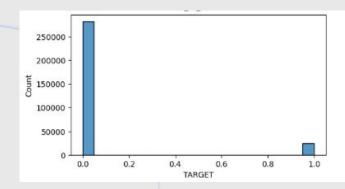
Target (whether individuals defaulted)
Heavily skewed - more 0s than 1s

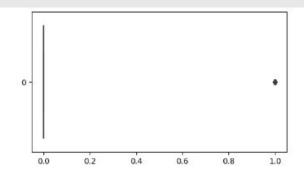
Repercussions:

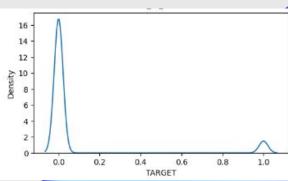
- 1. Biased model
- 2. Poor performance of minority class
- 3. Misleading evaluation metrics

Possible solutions:

- 1. Undersampling of majority class
- 2. Oversampling of minority class (SMOTE)
- 3. Cost-sensitive learning
- 4. Ensemble method

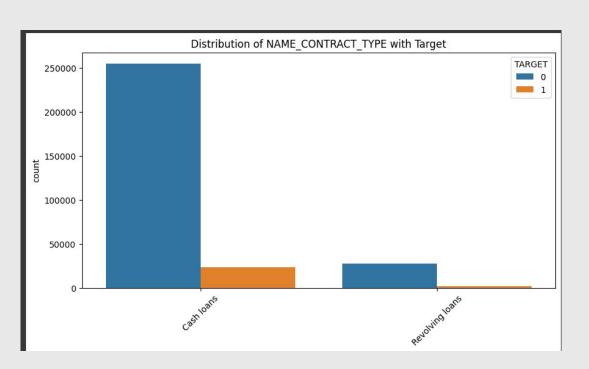






CATEGORICAL DATA

UNDERSAMPLING



CORRELATION (RAW DATA)



TARGET

DAYS EMPLOYED DAYS BIRTH

```
DAYS EMPLOYED
                           0.071695
DAYS BIRTH
                           0.063464
AMT GOODS PRICE
                          -0.029617
AMT INCOME TOTAL
                          -0.024722
AMT CREDIT
                          -0.020198
AMT ANNUITY
                           0.015305
CNT CHILDREN
                           0.010850
OBS 60 CNT SOCIAL CIRCLE
                          -0.009135
SK ID CURR
                           0.007665
OBS 30 CNT SOCIAL CIRCLE -0.007648
DEF 30 CNT SOCIAL CIRCLE
                                NaN
DEF 60 CNT SOCIAL CIRCLE
                                NaN
```

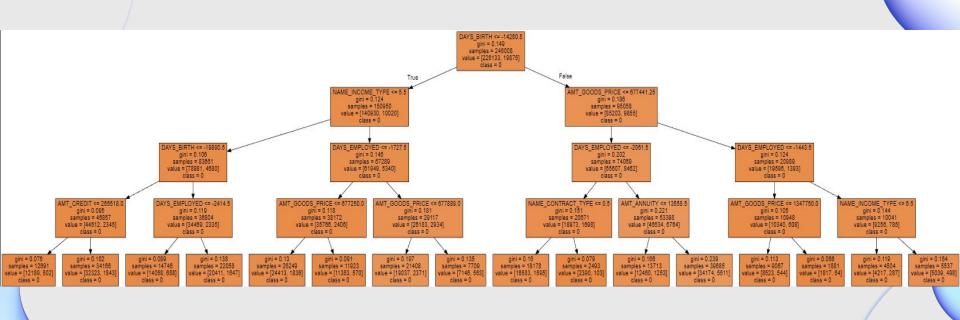
Name: TARGET, dtype: float64

The most correlated factor with TARGET is 'DAYS_EMPLOYED', with a correlation value of 0.0717.

Number of frauds (TARGET = 1) after dropping outliers: 1597

HACHINE SERNING

DECISION TREE



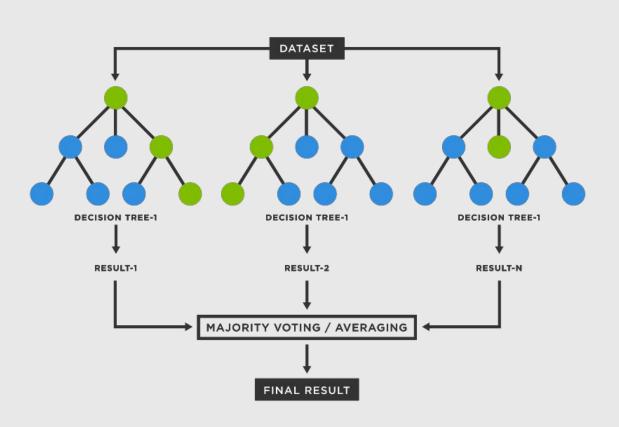
DECISION TREE

Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
91.92%	91.95%	0	0	0

	Predict Safe (0)	Predict Fraud (1)
Actual Safe (0)	56553	0
Actual Fraud (1)	4950	0

True Positive Rate	0%
False Positive Rate	0%

RANDOM FOREST



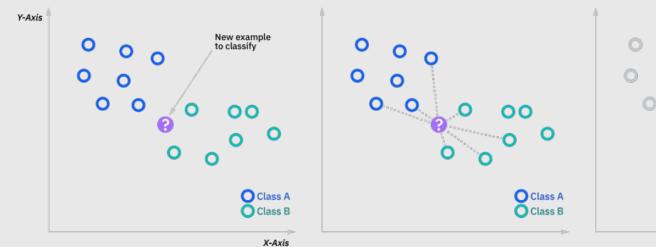
RANDOM FOREST

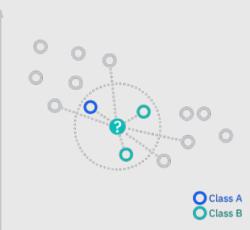
Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
98.40%	91.87%	0.2298	0.0040	0.0079

	Predict Safe (0)	Predict Fraud (1)
Actual Safe (0)	56486	67
Actual Fraud (1)	4930	20

True Positive Rate	0.40%
False Positive Rate	0.11%

K-NEAREST NEIGHBORS





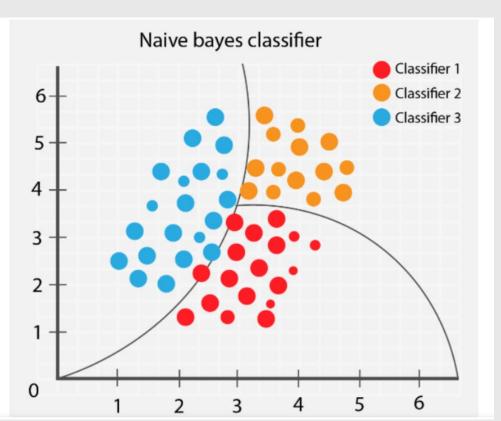
K-NEAREST NEIGHBORS

Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
92.70%	90.29%	0.1308	0.0365	0.0571

	Predict Safe (0)	Predict Fraud (1)
Actual Safe (0)	55351	1202
Actual Fraud (1)	4769	181

True Positive Rate	3.65%
False Positive Rate	2.12%

NAIVE BAYES



NAIVE BAYES

Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
91.30%	91.40%	0.0687	0.0054	0.0101

	Predict Safe (0)	Predict Fraud (1)
Actual Safe (0)	56187	366
Actual Fraud (1)	4923	27

True Positive Rate	0.54%
False Positive Rate	0.64%

MODEL COMPARISON

	model	train_acc	test_acc	precision	recall	f1_score
1	Decision Tree	0.919210	0.919516	0.000000	0.000000	0.000000
2	Random Forest	0.984074	0.918752	0.229885	0.004040	0.007941
3	K-Nearest Neighbours	0.927096	0.902915	0.130875	0.036566	0.057161
4	Gaussian Naive Bayes	0.913023	0.914004	0.068702	0.005455	0.010107

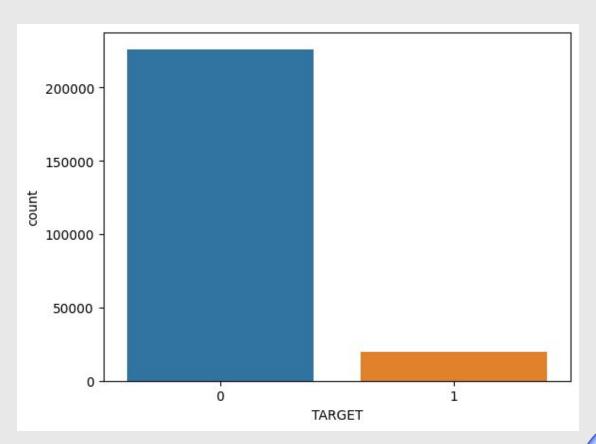
MODEL COMPARISON

- Highest test accuracy: Decision Tree Classifier
 - o **91.95%**
 - Predicts all customers as safe (target=0)
 - Useless at detecting if customer will default

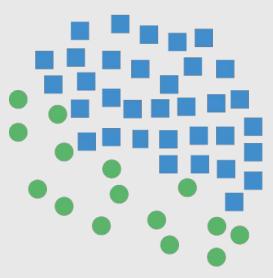


- Highest Recall (TPR): K-Nearest Neighbors
 - o **3.65%**
 - Extremely low detection rate
 - Takes extremely long time to predict

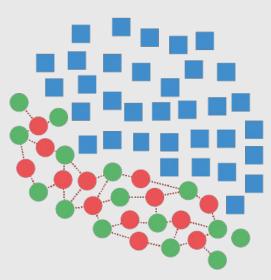
IMBALANCED TRAINING DATASET



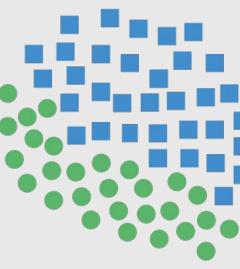
SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE)



Original Dataset

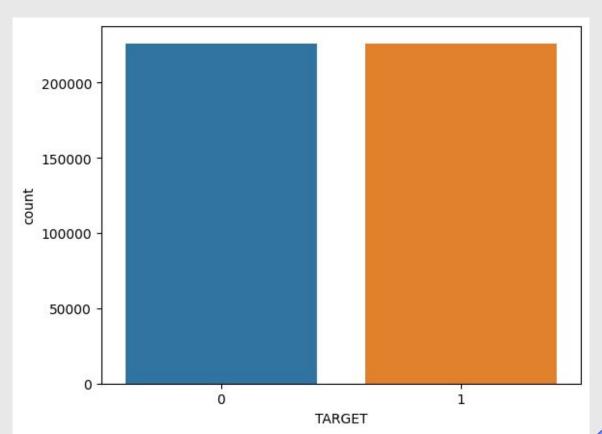


Generating Samples



Resampled Dataset

RESAMPLED TRAINING DATASET





CORRELATION (SMOTE)



TARGET

FLAG OWN CAR FLAG OWN REALTY -0.210410 DAYS EMPLOYED 0.142916 NAME CONTRACT TYPE -0.127142 DAYS BIRTH 0.124180 AMT GOODS PRICE -0.065508 AMT INCOME TOTAL -0.051444 AMT CREDIT -0.044920 NAME INCOME TYPE 0.027394 OCCUPATION TYPE -0.023302 OBS 60 CNT SOCIAL CIRCLE -0.019977 OBS 30 CNT SOCIAL CIRCLE -0.018194 NAME TYPE SUITE -0.016024 CNT CHILDREN 0.008865 AMT ANNUITY 0.003912 SK ID CURR 0.001456 DEF 30 CNT SOCIAL CIRCLE NaN DEF 60 CNT SOCIAL CIRCLE NaN Name: TARGET, dtype: float64

-9.278647

The most correlated factor with TARGET is 'FLAG OWN CAR', with a correlation value of -0.2786.

OWNING A FLAT OWNING A CAR

Model Comparison - SMOTE

Rank	model	train_acc	test_acc	precision	recall	f1_score
4	Decision Tree	0.725909	0.692487	0.089975	0.309495	0.139418
1	Random Forest	0.993763	0.897013	0.128755	0.048485	0.070443
3	K-Nearest Neighbours	0.926815	0.724013	0.097429	0.293939	0.146349
2	Gaussian Naive Bayes	0.497342	0.893339	0.063922	0.023838	0.034726

Model Comparison - SMOTE

- Sacrifice accuracy to improve model robustness
 - Decreased Accuracy
 - Increased Recall / F1 Score
- Highest test accuracy: Random Forest Classifier
 - 89.70% Accuracy
 - Low recall: 4.84%
- Highest Recall (TPR): Decision Tree Classifier
 - **30.94% Recall**
 - Low accuracy: 69.24%

Model Comparison - SMOTE

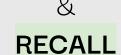
	model	train_acc	test_acc	precision	recall	f1_score
3	K-Nearest Neighbours	0.926815	0.724013	0.097429	0.293939	0.146349

BEST OVERALL MODEL: K-Nearest Neighbours











INSIGHTS & TAKEAWAYS

EDA

Understand overview of data



FEATURE SELECTION

Reduce training time



Model performance not affected



SMOTE

REDUCES ACCURACY



BUT

INCREASES PERFORMANCE

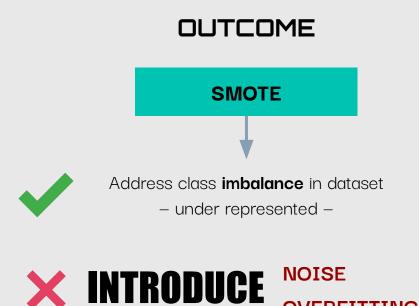


RECOMMENDATIONS

INSIGHTS & RECOMMENDATIONS

Contributing Factors

DAYS_EMPLOYED & DAYS_BIRTH



OVERFITTING

slidesgo