FRAUD DETECTION

Kool Data Kids
Rehearsing

Kool Data Kids

An efficient, cross-skilled and motivated team (only number 10)

CYPRIEN

Veepee Premium Member
Agile Slidemaster

ARNAUD

Applied ML research
Associate @ Harvard &
Debugging Master

TRISTAN

Wanna-be
surfer-rider-ascendant
snowboarder

MATHIEU

WoW champion
Velib Business Angel

XAVIER

“Dix pour cent” earliest fan
Lyft top 10 bike rider in SF

TABLE OF CONTENTS

01
PROBLEM
Fraudulent transactions are painful for your business

02
PROJECT OVERVIEW
From human detection to automation

03
READ THE DATA
There is very few frauds compared to the number of transaction every day

04
OUR APPROACH
Machine Learning can be used to detect frauds

05
NEXT STEPS
Recommendations to impact your business and implementation of our solution

01

PROBLEM

FRAUDULENT TRANSACTIONS ARE PAINFUL FOR YOUR BUSINESS

MONEY LOSS

$11

Million per year

CUSTOMER EXPERIENCE

Have your money stolen sucks

JUSTICE

Alignment with your values and purpose as a company

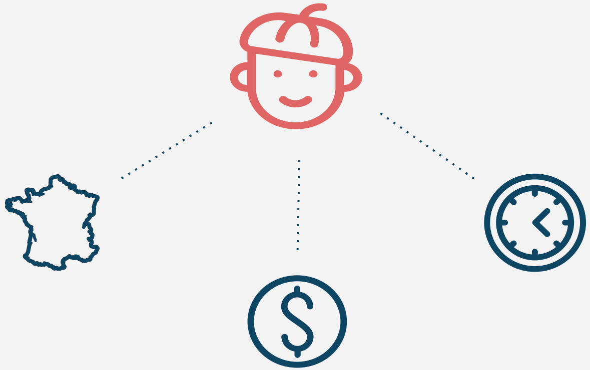
POLITICAL CREDIBILITY

Your reputation with regulation authorities is at stake

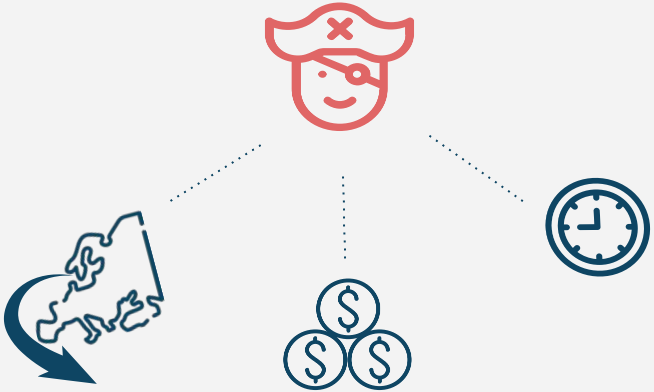
02

PROJECT OVERVIEW

Suspicious activity can easily be detected by humans



Typical transactions



Fraudulent transactions

51m transactions per year

Need for automation

1st pitfall

A fraud flies beneath the radar

The fraud is not detected and the transaction order is accepted

The credit card needs to be replaced, the client reassured and (eventually) refunded

2nd pitfall

A normal transaction is labeled as fraud by mistake

The client credit card is blocked on an unfounded suspicion of fraud

The client cannot use his/her card properly.

The credit card needs to be reactivated, the client reassured

We need to avoid **mislabeling**

03

READ THE DATA

An Imbalanced Dataset

99.83

percent of non-fraud transactions

0.17

percent of fraud transactions

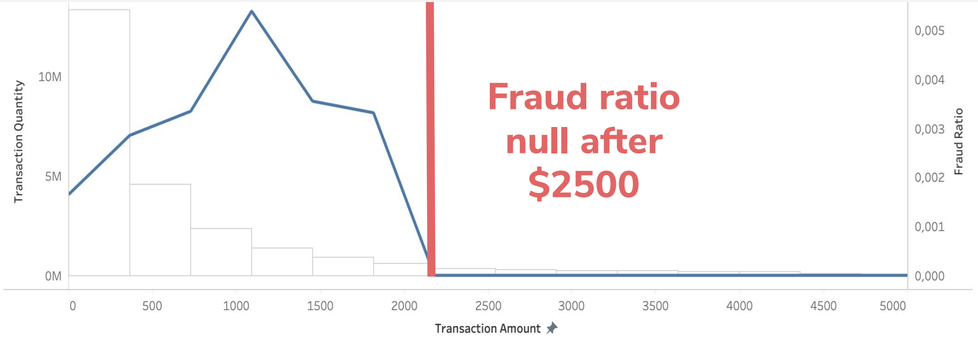
IN DEPTH

Fraud appear under specific circumstance

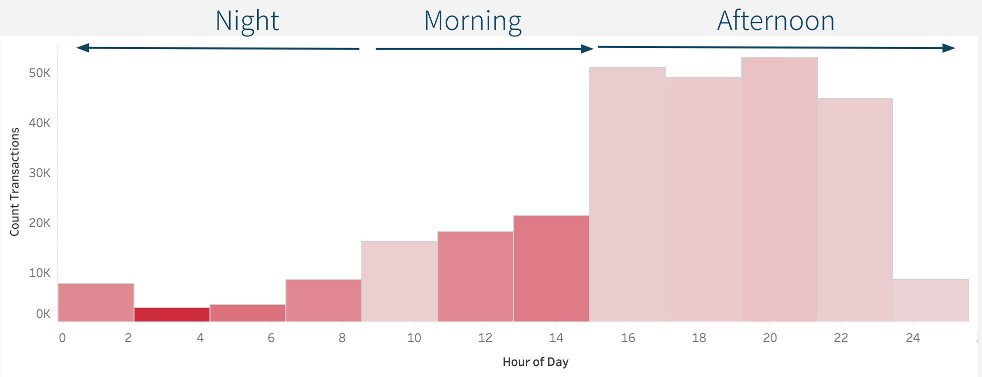
Over 31 columns, 3 were visible.

With more information we could find out more banking rules.

Industry rule prevents fraud higher than $2500?



Higher fraud risk at night and in the morning



04

OUR APPROACH

GREEDY MODEL

DATA PREPARATION

Standardization
Train / test split

CLASSIFIER

Train a model to predict the class

	Fraud	No Fraud
Labelled as Fraud	95 (Fraud Detected)	14 (Real Transaction Labelled as Fraud)
Labelled as Safe	54 (Fraud Not Detected)	93 824 (Validation of Real Transaction)

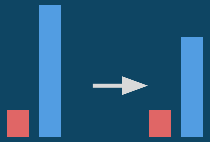
PIPELINE

DATA PREPARATION

Standardization
Train / test split

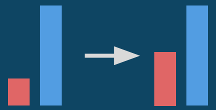
UNDERSAMPLING

Eliminate observation from the majority class



OVERSAMPLING

Create observation of the minority class

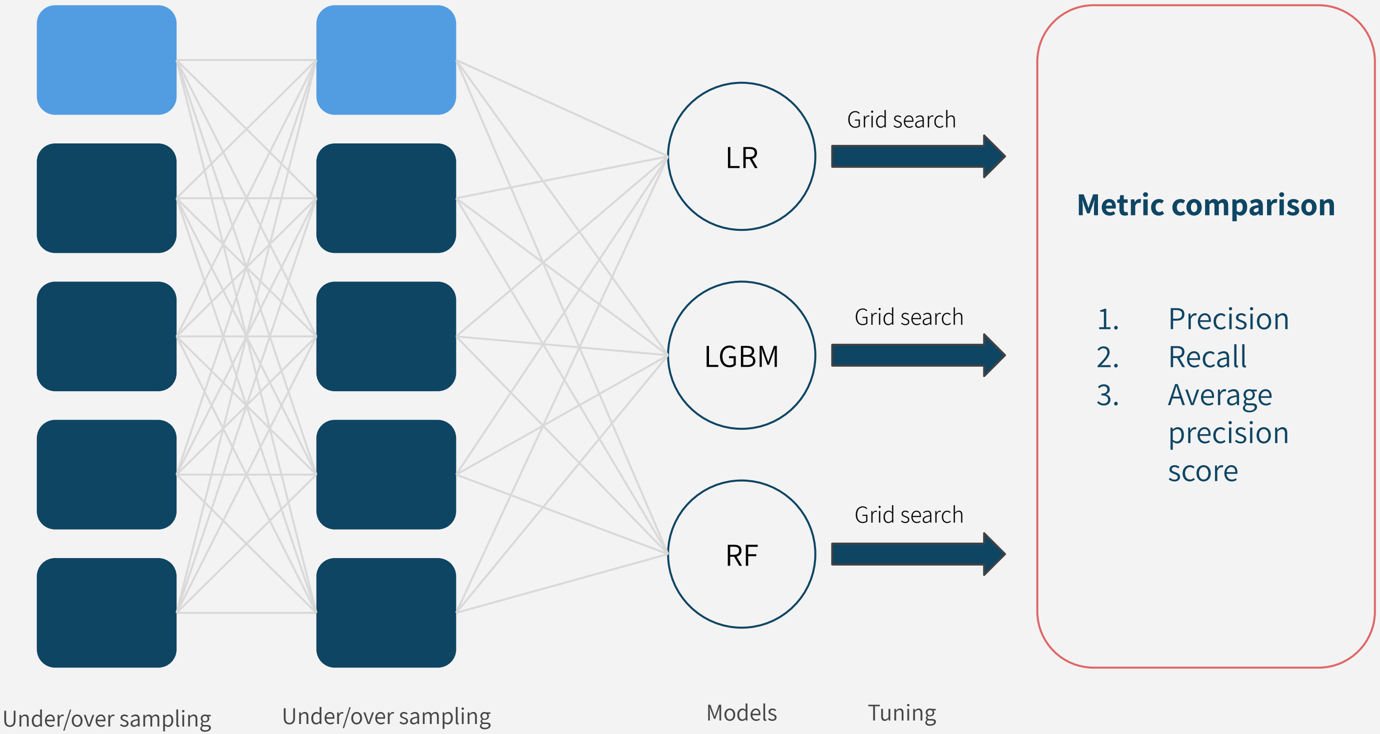


ALGORITHM MODIFICATION

- Weights
- Thresholds
- Custom Loss Function

Train a model to predict the class

PIPELINE SELECTION



+640

tested pipelines

8

CPU used

15h

of computation

OUR RESULTS

	Fraud	No Fraud
Labelled as Fraud	127 (Fraud Detected)	19 (Real Transaction Labelled as Fraud)
Labelled as Safe	22 (Fraud Not Detected)	93819 (Validation of Real Transaction)

KEY METRICS

85% of frauds are detected

0.02% of normal transactions are wrongly labeled

IMPACT ON YOUR BUSINESS

$ 900k saved per year

OUR RESULTS

	Fraud	No Fraud
Labelled as Fraud	+32 (Fraud Detected)	+5 (Real Transaction Labelled as Fraud)
Labelled as Safe	-32 (Fraud Not Detected)	-5 (Validation of Real Transaction)

KEY METRICS

85%
of frauds are detected

0.02%
of normal transactions are wrongly labeled

IMPACT ON YOUR BUSINESS

$ 900k saved per year

05

NEXT STEPS

NEXT STEPS

Define cost of operation to refine our custom loss function

#1 EVALUATE COSTS

#2 VALIDATION TESTING

A/B testing phase for 1 month

Implementing the solution in day-to-day operations

#3 PRODUCTION

#4 LEVEL OF RISK

Deal with fraud differently regarding the level of risk

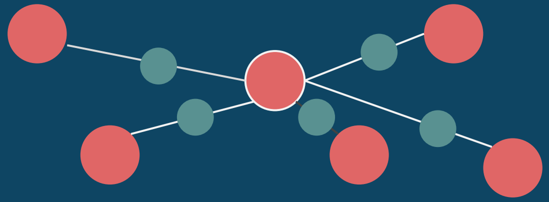
Automating fraud detection pipeline

#5 AUTOMATION

APPENDIX 1: One Sided Selection

APPENDIX 2: SMOTE
(Synthetic Minority Oversampling Technique)

SMOTE creates new minority observations between existing minority observations .

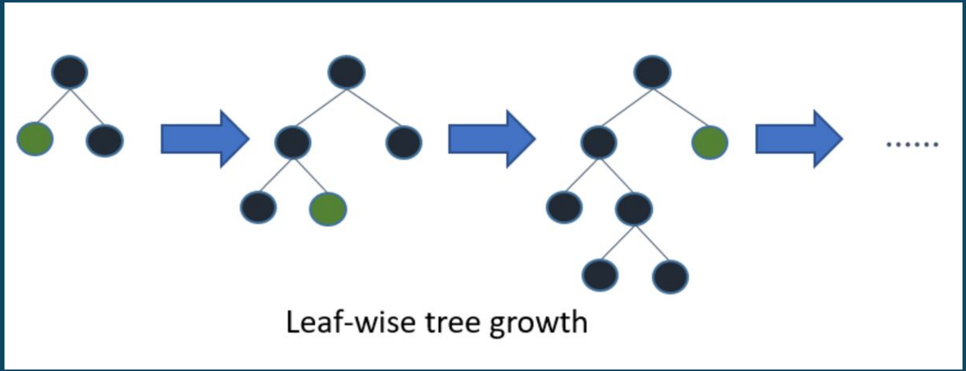


- For each minority instance, k nearest neighbors of the same class are found
- The difference between the feature vector of the considered observation and the feature vectors of the k nearest neighbors are found. k difference vectors are obtained
- Each on of the k difference vector is multiplied by a random number between 0 and 1 (excluding 0 and 1).
- Then they are added to the feature vector of the considered observation at each iteration

APPENDIX 3: Random Forest

APPENDIX 4: Light GBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm.



Light GBM is prefixed as ‘Light’ because of its high speed. Light GBM can handle the large size of data and takes lower memory to run.

APPENDIX 5: Tested pipelines

	accuracy	precision	recall	f1_score	specificity_score	geometric_mean_score	average_precision_score
LogisticRegression_NoneType	0.9992765	0.9992221	0.9992765	0.9992205	0.638158204	0.798559011	0.556266931
LogisticRegression_NeighbourhoodCleaningRule	0.9992871	0.9992351	0.9992871	0.9992398	0.658260479	0.811043296	0.565949434
LogisticRegression_NearMiss	0.9991914	0.9991209	0.9991914	0.9991139	0.584552047	0.764250852	0.503616673
LogisticRegression_EditedNearestNeighbours	0.9990743	0.999117	0.9990743	0.9990937	0.752070679	0.866818619	0.520072187
LogisticRegression_ClusterCentroids	0.9994042	0.9993729	0.9994042	0.9993805	0.738669697	0.859202874	0.639848445
LogisticRegression_CondensedNearestNeighbour	0.9993829	0.9993516	0.9993829	0.9993607	0.738669664	0.859193707	0.629934747
LogisticRegression_AllKNN	0.999202	0.9991336	0.999202	0.9991385	0.611355074	0.781579953	0.515220636
LogisticRegression_InstanceHardnessThreshold	0.9994787	0.9994705	0.9994787	0.9994742	0.819078848	0.904793801	0.693986006
LogisticRegression_OneSidedSelection	0.6208093	0.9982665	0.6208093	0.7644163	0.93909113	0.763542055	0.003773243
LogisticRegression_RandomUnderSampler	0.5621097	0.9982685	0.5621097	0.7180114	0.945698677	0.729099688	0.003316718
LogisticRegression_TomekLinks	0.7372722	0.9981922	0.7372722	0.8471949	0.892370786	0.811122772	0.004955328
LinearSVC_NoneType	0.2849543	0.9981932	0.2849543	0.4415091	0.958660103	0.522660809	0.002101818
LinearSVC_NeighbourhoodCleaningRule	0.8724824	0.998274	0.8724824	0.9303517	0.912687737	0.892358647	0.010390566
LinearSVC_NearMiss	0.6525264	0.9982419	0.6525264	0.7881195	0.925739986	0.77721927	0.00401553
LinearSVC_EditedNearestNeighbours	0.8653537	0.9981849	0.8653537	0.9262719	0.865771148	0.865562409	0.008962761
LinearSVC_ClusterCentroids	0.636886	0.9982037	0.636886	0.7765544	0.912313646	0.762259631	0.003762568
LinearSVC_CondensedNearestNeighbour	0.8717908	0.9982861	0.8717908	0.9299563	0.919387391	0.895272836	0.010473997
LinearSVC_AllKNN	0.6525264	0.9982419	0.6525264	0.7881195	0.925739986	0.77721927	0.00401553
LinearSVC_InstanceHardnessThreshold	0.8758977	0.9981892	0.8758977	0.9323009	0.86578789	0.870828139	0.009699295
LinearSVC_OneSidedSelection	0.6330769	0.9982023	0.6330769	0.773703	0.912307598	0.759974263	0.003725078
LinearSVC_RandomUnderSampler	0.9334376	0.9982189	0.9334376	0.9640562	0.865879255	0.899024058	0.017759467
LinearSVC_TomekLinks	0.680924	0.9980776	0.680924	0.8086019	0.852076797	0.761708281	0.003831082
RandomForestClassifier_NoneType	0.9074872	0.9982035	0.9074872	0.9499676	0.86583805	0.886418046	0.012898457
RandomForestClassifier_NeighbourhoodCleaningRul	0.8880909	0.9982065	0.8880909	0.939189	0.872508004	0.880264974	0.010873227
RandomForestClassifier_NearMiss	0.9334376	0.9982189	0.9334376	0.9640562	0.865879255	0.899024058	0.017759467
RandomForestClassifier_EditedNearestNeighbours	0.680924	0.9980776	0.680924	0.8086019	0.852076797	0.761708281	0.003831082
RandomForestClassifier_ClusterCentroids	0.9074872	0.9982035	0.9074872	0.9499676	0.86583805	0.886418046	0.012898457
RandomForestClassifier_CondensedNearestNeighbou	0.8880909	0.9982065	0.8880909	0.939189	0.872508004	0.880264974	0.010873227
RandomForestClassifier_AllKNN	0.9266069	0.998226	0.9266069	0.9603835	0.872569161	0.899182173	0.016384609
RandomForestClassifier_InstanceHardnessThreshold	0.6063817	0.997911	0.6063817	0.7533898	0.805053167	0.698691302	0.002914504
RandomForestClassifier_OneSidedSelection	0.8651622	0.998222	0.8651622	0.92616	0.885873102	0.87545641	0.009325355
RandomForestClassifier_RandomUnderSampler	0.6868078	0.9982205	0.6868078	0.8127334	0.912392914	0.791605031	0.004337858
RandomForestClassifier_TomekLinks	0.796291	0.998279	0.796291	0.8850262	0.925968262	0.858685141	0.006748703
LGBMClassifier_NoneType	0.5450967	0.9982244	0.5450967	0.7039105	0.932270158	0.712865591	0.003130163
LGBMClassifier_NeighbourhoodCleaningRule	0.254099	0.9982497	0.254099	0.403129	0.972012615	0.496978275	0.002051319
LGBMClassifier_NearMiss	0.7300584	0.9969492	0.7300584	0.8425456	0.322795349	0.48544769	0.001685371
LGBMClassifier_EditedNearestNeighbours	0.5316161	0.9982394	0.5316161	0.6925042	0.938949505	0.706513022	0.003075026
LGBMClassifier_ClusterCentroids	0.5093258	0.9982312	0.5093258	0.6732027	0.938914112	0.691529567	0.002940067
LGBMClassifier_CondensedNearestNeighbour	0.03376	0.9977578	0.03376	0.0623841	0.985064257	0.182361612	0.0016157
LGBMClassifier_AllKNN	0.9984147	0.9968319	0.9984147	0.9976226	0.001585326	0.039784575	0.001585326
LGBMClassifier_InstanceHardnessThreshold	0.5316161	0.9982394	0.5316161	0.6925042	0.938949505	0.706513022	0.003075026
LGBMClassifier_OneSidedSelection	0.5093683	0.9982313	0.5093683	0.6732401	0.938914179	0.691558483	0.002940313
LGBMClassifier_RandomUnderSampler	0.03376	0.9977578	0.03376	0.0623841	0.985064257	0.182361612	0.0016157
LGBMClassifier_TomekLinks	0.9984147	0.9968319	0.9984147	0.9976226	0.001585326	0.039784575	0.001585326
LogisticRegression_NoneType	0.9992765	0.9992221	0.9992765	0.9992205	0.638158204	0.798559011	0.556266931
LogisticRegression_SMOTE	0.9991914	0.9991209	0.9991914	0.9991139	0.584552047	0.764250852	0.503616673
LogisticRegression_BorderlineSMOTE	0.999202	0.9991706	0.999202	0.9991838	0.705165613	0.83940628	0.544536448
LogisticRegression_RandomOverSampler	0.9747305	0.9983591	0.9747305	0.9857827	0.912850091	0.943283027	0.049831763
LogisticRegression_SVMSMOTE	0.9959462	0.998655	0.9959462	0.9970329	0.886080767	0.939408758	0.235946193
RandomForestClassifier_NoneType	0.999468	0.9994645	0.999468	0.9994662	0.825779584	0.908482404	0.691003932
RandomForestClassifier_SMOTE	0.9894453	0.9983854	0.9894453	0.9934377	0.865968186	0.925650147	0.101652
RandomForestClassifier_BorderlineSMOTE	0.989041	0.9984264	0.989041	0.9932313	0.892770555	0.939673729	0.103673393
RandomForestClassifier_RandomOverSampler	0.9996702	0.999661	0.9996702	0.9996627	0.852582916	0.923201877	0.79617762
RandomForestClassifier_SVMSMOTE	0.975337	0.9983612	0.975337	0.9860972	0.912851054	0.943576929	0.050992159
LGBMClassifier_NoneType	0.9962442	0.9986803	0.9962442	0.9972119	0.88608124	0.939549499	0.250051811
LGBMClassifier_SMOTE	0.9996382	0.9996273	0.9996382	0.9996293	0.83918136	0.915902716	0.777038849
LGBMClassifier_BorderlineSMOTE	0.991435	0.9984717	0.991435	0.9945082	0.892774356	0.940812272	0.128931756
LGBMClassifier_RandomOverSampler	0.9919138	0.9984715	0.9919138	0.9947646	0.886074364	0.937501663	0.13382613
LGBMClassifier_SVMSMOTE	0.9995744	0.9995717	0.9995744	0.999573	0.859283517	0.92677819	0.74824882
LogisticRegression_SMOTEENN	0.9734538	0.998355	0.9734538	0.9851205	0.912848063	0.942663988	0.047553989
LogisticRegression_SMOTETomek	0.9781459	0.9983723	0.9781459	0.9875537	0.912855514	0.944936977	0.057159092
LinearSVC_SMOTEENN	0.9959462	0.998655	0.9959462	0.9970329	0.886080767	0.939408758	0.235946193
LinearSVC_SMOTETomek	0.9992765	0.9993638	0.9992765	0.9993095	0.872684549	0.933837865	0.633849188
RandomForestClassifier_SMOTEENN	0.973507	0.9983551	0.973507	0.9851481	0.912848148	0.942689789	0.047644719
RandomForestClassifier_SMOTETomek	0.9787417	0.998375	0.9787417	0.9878626	0.91285646	0.945225222	0.058664591
LGBMClassifier_SMOTEENN	0.9959462	0.998655	0.9959462	0.9970329	0.886080767	0.939408758	0.235946193
LGBMClassifier_SMOTETomek	0.9993084	0.9993898	0.9993084	0.999339	0.879385353	0.937431162	0.64723925

APPENDIX 6: THE MAGIC OF KEDRO

CREDIT

To create this presentation, Slidesgo was used (Credit)