Signal Detection in Noisy Financial Data

Finding Needles in Haystacks

From credit default events to trade executions, imbalanced datasets are pervasive in finance. Detecting signal in such noisy data presents well documented challenges. This talk is a practitioner's point of view on using novel techniques for probability estimation in the ML context after under-sampling techniques.

Takeaways: A Preview

Executive

- Transition knowledge of traditional financial modeling towards more powerful ML tools
- Data driven idea generation

Trader

- Improve product specification and requirements
- Tools to challenge your quant

Data Scientist

- Decompose your ML model to improve importance and interpretability
- Adjust your probabilities after sampling

Examples of imbalanced datasets in Finance

Types	Sample size	Event size
Credit Defaults	# Companies: ~1-10k	# Defaults: 1-10%
Consumer Defaults	# Consumers: ~10-100M	# Non-payments: 1-10%
Fraud Detection	# Transactions: ~1B / day	# Fraud Events: <1%
Trade Executions	# RFQs: widely varying	# Executions: <1%
Trade Recommendations	# Securities: ~10k	# Recommendations: <1%

The Problem: Stating the problem

Predicting class **isn't enough**, despite this being the ask.
There are massive gains by aiming a **step deeper**.

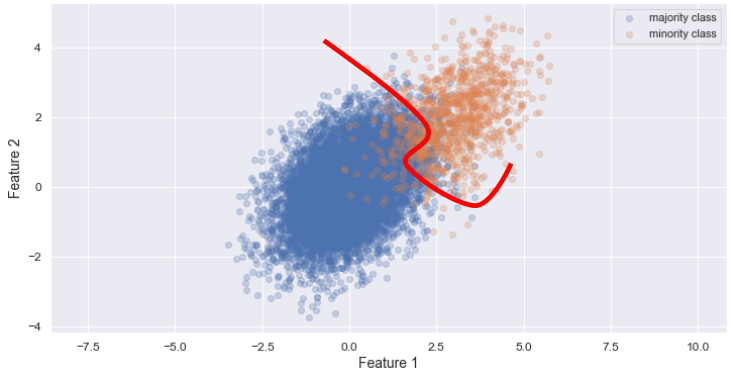
Default? → $ Loss Conditional on Default

Recommend?→ Average $ Spend if Recommend

Common Objectives in Event Detection

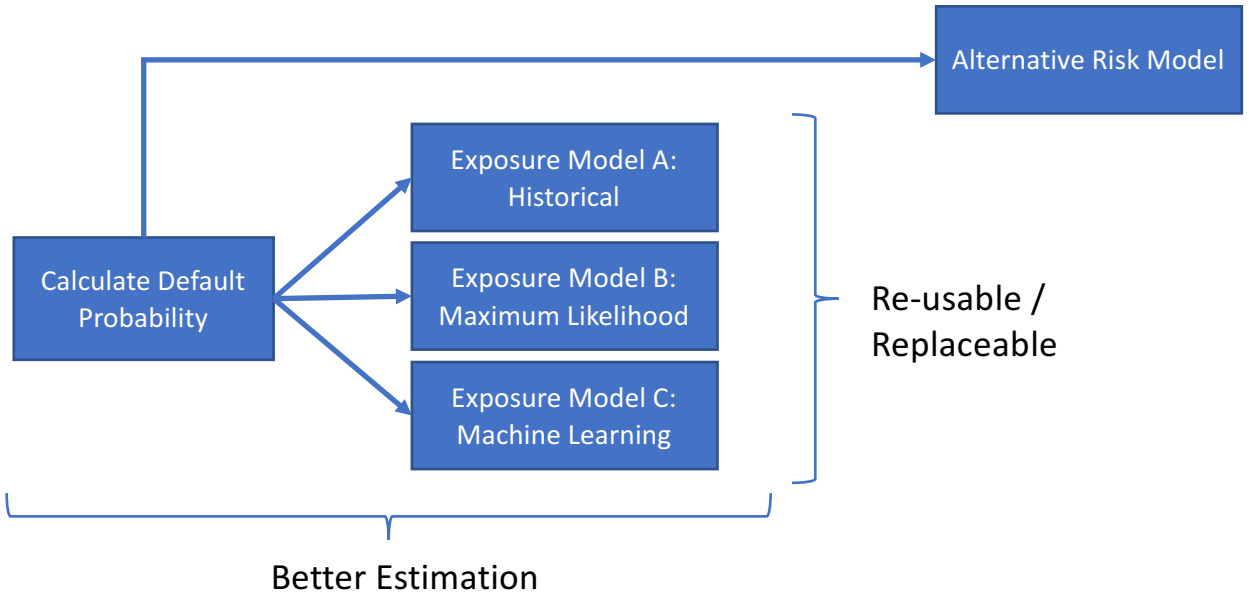
Identify class separating hypercube: 0 or 1

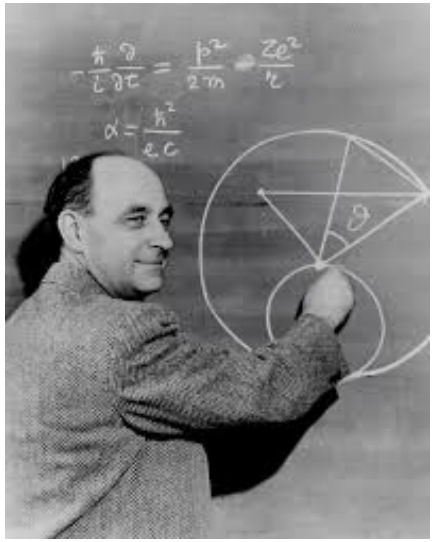
Calculate the probability



Modularize Your Models

- **Better Estimation**: Break calculations into a series of smaller components: **Fermi Estimation**
- **Model Hygiene: Re-usable** and **replaceable** model components



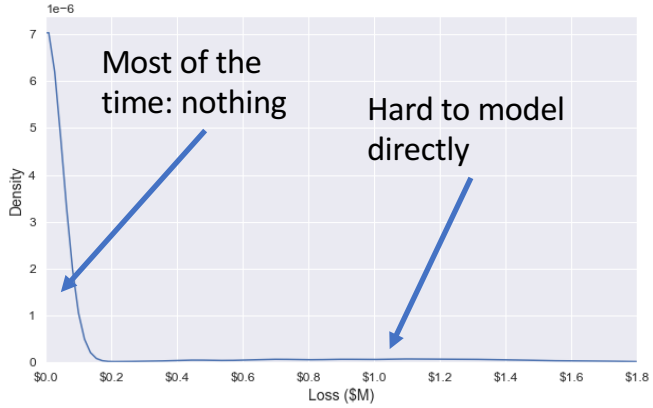


Wiki: Fermi Estimation

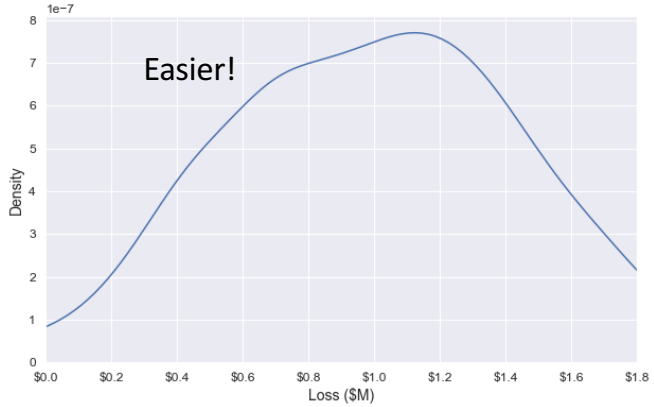
Standard credit modeling (above) can be applied to other imbalanced data problems

"Standard Trick" in Financial Modeling: Requires Probability!

Unconditional Loss Distribution



Conditional Loss Distribution



$$E[L] = E[L|D = 1]P(D = 1)$$ [See Appendix]

- This approach can be applied to many types of imbalanced / noisy problems
- New Data Scientists try to model unconditional directly (don't!)
- Experienced non-ML quants have myriad techniques to handle (see importance sampling)
- ML models typically use sampling in imbalanced data sets, but this distorts probability: need an adjustment!

Simulation

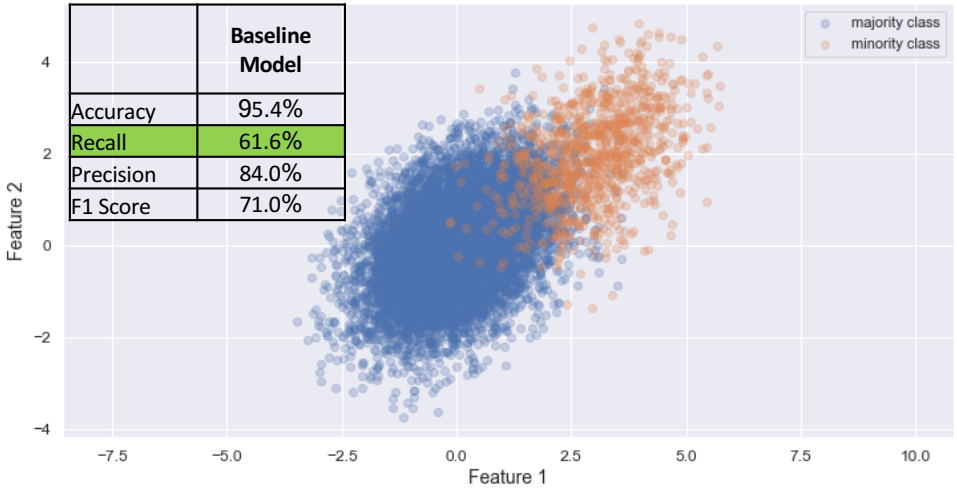
Simulated Imbalance	Under-sample Ratio	Beta Adjustment Factor	Actual Probability	Under-Sample Probability	Under-Sample Adjusted Probability
0.95	10	0.53	5%	9%	5%
0.99	10	0.10	1%	9%	1%
0.995	10	0.05	0.5%	9%	0.5%

A Toy Example

Types	Example Features
Credit Defaults	Stock price, Average sector spread
Consumer Defaults	Credit score, No. late payments last year
Fraud Detection	Ratio of $ transaction last month to $ transactions yesterday, Transaction size
Trade Executions	Historical execution ratios, Market volume

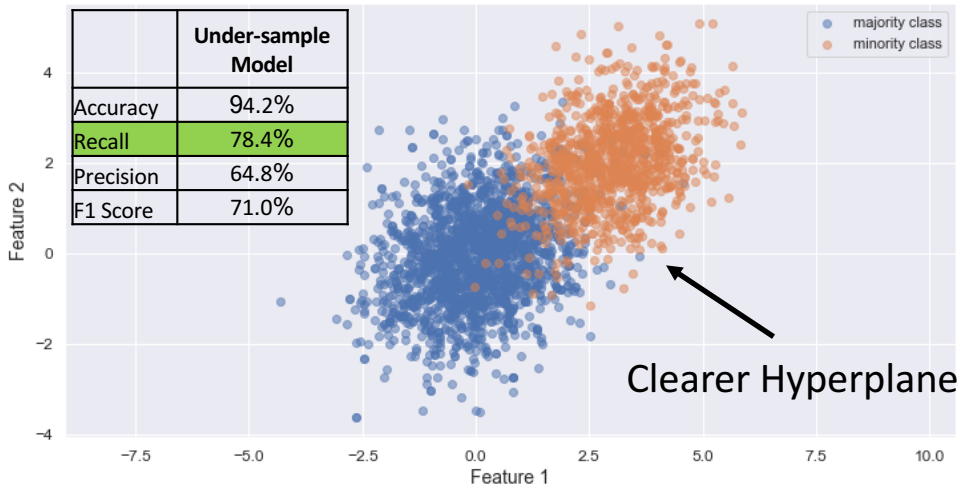
Original

	Baseline Model
Accuracy	95.4%
Recall	61.6%
Precision	84.0%
F1 Score	71.0%



Under-sampled Majority Class

	Under-sample Model
Accuracy	94.2%
Recall	78.4%
Precision	64.8%
F1 Score	71.0%



Analysis Details

- Sample data: 2 dimensional multi-variate normal
- Random under-sampling, 4:1 majority to minority
- Random Forest Classifier – max depth=7, estimators=100
- Metrics generally improve on real data
- Github link for code

Sampling Techniques[1]

Under-Sampling

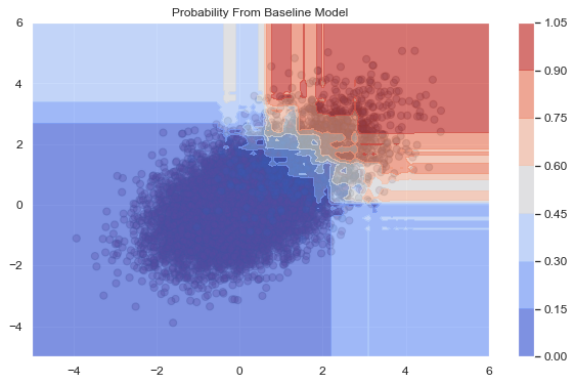
- Random Under-sample
- Near-Miss Algorithms
- Edited Nearest Neighbors
- Condensed Nearest Neighbor

Over-Sampling

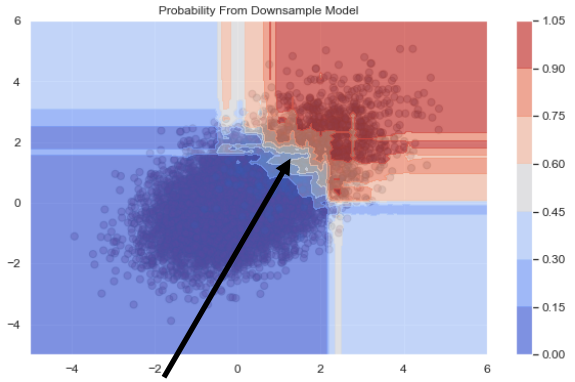
- Random Over-sample
- Random Over-Sampling Examples (ROSE)
- Synthetic Minority Over-sampling Technique (SMOTE)
- Adaptive Synthetic Sampling (ADASYN)

[1] Imblearn Library

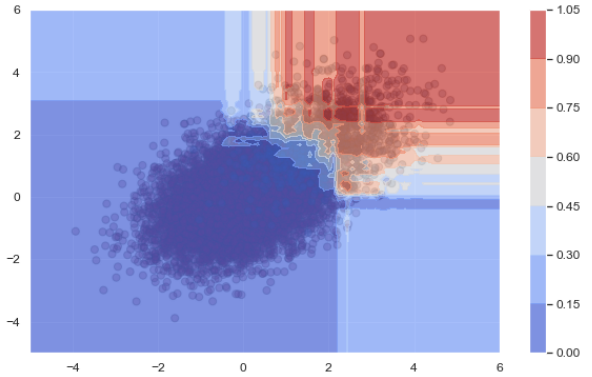
No Sampling



Under-Sampling



Under-Sampling w Adjustment



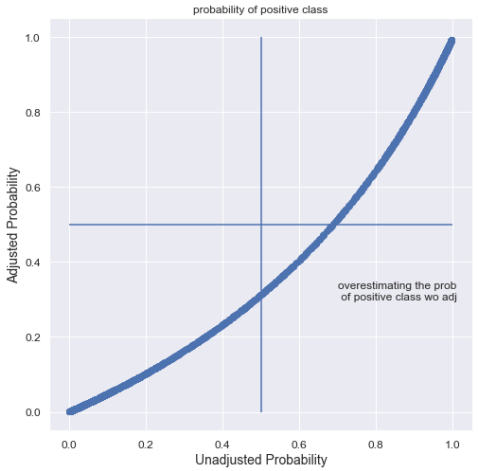
More nuanced probability contours around hyperplane, but 'wrong' probabilities.
cls.predict_probaba(*) fails!

Novel Adjustment[1]

$$p = \frac{\beta p_s}{\beta p_s - p_s + 1}$$

[See Appendix for full derivation]

The Fix



[1]A. D. Pozzolo, O. Caelen, R. A. Johnson and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159-166, doi: 10.1109/SSCI.2015.33.

Takeaways: A Recap

- Drive a more **modular** approach to Machine Learning solutions.
- Specify the **final** desired outcome.
- In the modular paradigm, **probability is more desirable** than class predictions.
- In noisy data, this typically requires **sampling**, which will mean it will **always** require probability adjustments.

Appendix: Calculations

"Standard Trick"

- In some cases, predicting class is enough. Often, massive modeling gains by estimating the conditional distribution rather than the unconditional distribution.

$$E[L] = \int_{\Omega} L(\omega) dF(\omega)$$

$$E[L] = \int_{\Omega} L(\omega) 1(\omega)_{\{D=1\}} dF(\omega) + \int_{\Omega} L(\omega) 1(\omega)_{\{D=0\}} dF(\omega)$$

$$E[L] = E[L|D=1]P(D=1)$$

Probability Adjustment[1]

s: sample included or not
Z: event occurs

Two abbreviations:
$P(Z=1)$: p
$P(Z=1|s=1)$: p_s

Defining $P(s=1)$:

$$P(s=1) = P(s=1|Z=1)P(Z=1) + P(s=1|Z=0)P(Z=0) = p + P(s=1|Z=0)(1-p)$$

$$\beta = P(s=1|Z=0)$$

$$P(s=1) = p + \beta(1-p)$$

Reducing relationship:

$$P(Z=1|s=1)P(s=1) = P(s=1|Z=1)P(Z=1)$$

$P(s=1|Z=1) = 1$ minority class is always included

$$P(Z=1|s=1)P(s=1) = P(Z=1)$$

$$p_s(p + \beta(1-p)) = p$$

$$p = \frac{\beta p_s}{\beta p_s - p_s + 1}$$

[1]A. D. Pozzolo, O. Caelen, R. A. Johnson and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159-166, doi: 10.1109/SSCI.2015.33.