＜ 修 士 論 文 ＞

不均衡データに対する機械学習手法と税関不正検知への応用

滋 賀 大 学 大 学 院
デ ー タ サ イ エ ン ス 研 究 科
デ ー タ サ イ エ ン ス 専 攻

修了年度 ： ２０２０年度

学籍番号 ： 6019114
氏　　名 ： 辻　　孝 辰
指導教員 ： 松井　秀俊

提出年月日 ： 2021年1月20日

目 次

第1章　はじめに

1.1　税関不正検知

税関では，「安全・安心な社会の実現」，「適正かつ公平な関税等の徴収」及び「貿易円滑化の推進」という三つの使命を掲げ，貿易の健全な発展と安全な社会の実現に努めている．近年，貿易額が増大しており，それに伴って税関の業務量は増加の一途を辿っている．また，訪日外国人旅行者数については，現在のところ新型コロナウイルスの流行により一時的に激減しているものの，流行前までは貿易額と同様に増加の一途を辿っていた．よって，流行が収まれば再び流行前の数に戻り，さらに増加し続けると考えられる．

そのような中，限られた人員で三つの使命を果たしていくためには，税関業務の高度化・効率化が必要不可欠である．そのため，税関ではデータサイエンス等の先端技術を活用するための試行・検証を積極的に推進している．例えば，通関審査・検査選定業務や輸入事後調査立入先選定業務等を支援するために，輸入申告等の膨大なデータを解析している．特に，解析によって不正な申告等を自動的に検知することを税関不正検知 (Customs fraud detection) と呼び，我が国だけでなく，各国の税関当局において様々な試みがなされている．従来の税関不正検知はルールベースによるものが主流であったが，近年はそれに代わり機械学習を活用したものが検討及び導入され始めている．機械学習を活用した税関不正検知の課題として，Kim らは解釈可能性，過去データの利用可能性，不正パターンの変化，ラベル付きデータの入手可能性，不均衡データ，プライバシーの 6 つを挙げている [1]．これらの中で，不均衡データは機械学習モデルの精度に悪影響を与えることが知られている．

1.2　本研究のねらい

機械学習を活用した税関不正検知において，不均衡データへの対処が課題の一つであるものの，Vanhoeyveld らは税関不正検知の先行研究において，この課題への考慮が欠けていると主張している [2]．実際に，現在でもこの問題の解決策を提案している論文は少ない．

さらに，税関不正検知において，不正のあった例が全体に占める割合は非常に小さい．つまり，その不均衡である度合いが非常に大きく，一般的な機械学習手法はほとんど役に立たない．以上より，不均衡問題について研究し，税関不正検知に効果的な手法を検討することには，税関ひいては国民にとって大きな意義があると考える．また，現在数ある不均衡問題の解決策について，それらを第三者の視点から公平に評価することは，学術的にも価値があるといえる．したがって，本研究では不均衡データに対する既存の機械学習手法を調査し，その中で税関不正検知に有効とされる手法について，改善策を検討し，その効果を検証する．

本論文の構成は次の通りである．第 2 章では，不均衡データに対する既存の機械学習手法を紹介し，それらを不均衡データ解析のベンチマークとして一般的に使用されているデータセットに適用することで評価する．第 3 章では，第 2 章で紹介した手法を，税関への輸入申告の疑似データと実データに対して適用し，その効果を検証する．第 4 章では，第 3 章で効果のあった手法に対する改善策を検討し，その効果を検証する．最後に，第 5 章でまとめと今後の課題について述べる．

第2章　不均衡データ解析に関する先行研究

不均衡データに対する既存の機械学習手法には大きく分けて，データに基づくアプローチとアルゴリズムに基づくアプローチがあり，さらにそれらとアンサンブル学習を組み合わせたものがある．本章では，それらのうち代表的なものを紹介する．なお，以降の説明では二値分類タスクを対象とすることとし，少数クラス (minor class) のサンプルを正例 (positive samples)，多数クラス (major class) のサンプルを負例 (negative samples) とする．

2.1　不均衡データ

まず，予備知識として，不均衡データが機械学習モデルに与える影響について紹介する．不均衡データとは，分類問題において各クラスのサンプルサイズが均衡でない，つまり，各クラスのサンプルサイズに偏りがあるデータのことをいう．例として，患者が癌化するかどうかを機械学習によって予測するタスクを考える．一般的に，癌化しなかった例に対して癌化した例は非常に少なく，訓練データにおける「癌化しなかった」クラスと「癌化した」クラスのサンプルサイズには大きな偏りがある．このような不均衡データによりモデルを訓練すると，癌化しないと予測しがちなモデルが構築される．これは，一般的な機械学習モデルが誤り率の最小化を目的として設計されているからである．例えば，訓練データのうち，癌化しなかった例が 990 例，癌化した例が 10 例である場合，全て癌化しないと予測するモデルの訓練誤差は 10/1000，つまり 1%となり，誤り率の観点では非常に良いモデルと評価される．しかし，そのようなモデルが全くの無価値であることは明らかである．なぜならば，このタスクにおいて，癌化する患者を癌化しないと予測するリスクは，癌化しない患者を癌化すると予測するリスクに比べて非常に大きいからである．このタスクと類似するタスクとしてよく挙げられるのは，金融業における貸し倒れ予測やスパムメール検知などである．

不均衡データに関する研究においては，データが不均衡である度合いを不均衡比率 (IR: Imbalance Ratio) で表すことが一般的である．二値分類タスクの場合，不均衡比率は多数

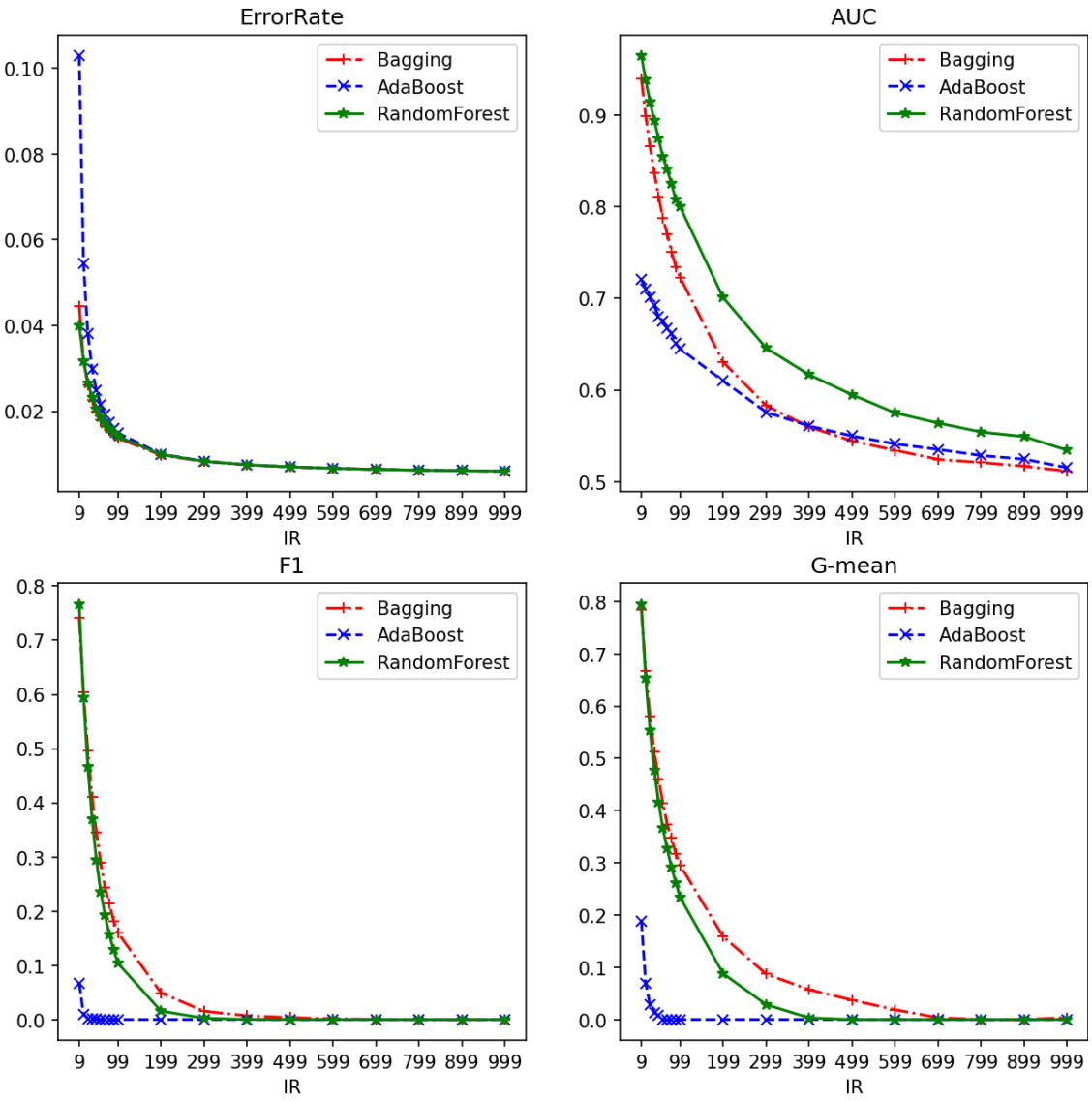

図 2.1: 不均衡比率とモデルの性能評価指標

クラスのサンプルサイズを少数クラスのそれで割った値であり，この値が大きいほどより不均衡であるといえる．データの不均衡比率を変化させたときのモデルの性能評価指標の変化を図 2.1 に示す．データは Python の scikit-learn[3] パッケージの make_classification 関数により，人工的に生成したものである．図の縦軸はそれぞれ，誤り率 (ErrorRate)，AUC，F1 値，G-mean である．AUC は ROC (Receiver Operating Characteristic) 曲線の下側面積 (Area Under Curve) である．F1 値は適合率 (Precision) と再現率 (Recall) の調和平均であり，次式により算出される．

$$\mathrm{F1} = 2\frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.1.1}$$

G-mean(Geometric mean) は真陽性率 (TPR: True Positive Rate) と真陰性率 (TNR: True

Negative Rate) の幾何平均であり，次式により算出される．

$$\text{G-mean} = \sqrt{\text{TPR} \times \text{TNR}} = \sqrt{\frac{\text{TP}}{\text{TP}+\text{FN}} \times \frac{\text{TN}}{\text{TN}+\text{FP}}} \tag{2.1.2}$$

機械学習モデルはバギング (Bagging)，アダブースト (AdaBoost)，ランダムフォレスト (RandomForest) の 3 つとした．まず，誤り率に注目すると，不均衡比率が高くなるにつれて減少することが確認できる．一見すると誤り率が低くて良いモデルが構築されているように見える．しかし，アダブーストでは不均衡比率が 9 の時に誤り率がおよそ 0.1 で，不均衡比率が 99 の時には誤り率がおよそ 0.01 である．これは，少数クラスのサンプルサイズが全サンプルサイズに占める割合とほぼ同じ値となっている．つまり，このモデルは先述の通りほぼ全てのサンプルを多数クラスに分類している可能性が高いといえる．その証拠として，アダブーストの F1 値と G-mean は著しく低く，不均衡比率が 99 を超えると，いずれもほぼ 0 となっている．G-mean がほぼ 0 ということは，少なくとも真陽性率又は真陰性率がほぼ 0 であることを意味するが，誤り率が非常に低いことより，いずれか一方は高いはずである．そして，F1 値が低いことは，正例に対する予測精度である適合率又は再現率が低いことを意味する．再現率は真陽性率と同義であることから，ここでは真陽性率がほぼ 0 であると考えられ，このモデルの予測は負例に大きく偏っているといえる．他の 2 つのモデルも不均衡比率が高くなるにつれて F1 値と G-mean が対数関数的に減少していくことが確認できる．なお，AUC についても，F1 値や G-mean に比べて減少が緩やかであるものの，不均衡比率が大きくなるにつれて最低値である 0.5 に収束している．

また，Prati らは分類が困難なデータであるほど，不均衡比率がモデルの精度へ与える影響が大きくなることを実験によって検証している [4]．この実験では，各クラスの重心同士の距離と不均衡比率をそれぞれ変化させて，前者が小さくなるほど，後者の増加に伴う AUC の低下が大きくなるという結果が得られている．

以上から，不均衡データは機械学習モデルの精度に悪影響を与え，また，分類が困難なデータほどその影響が大きくなるといえる．以降，不均衡データに対する既存の機械学習手法について紹介する．

2.2 データに基づくアプローチ

データに基づくアプローチでは，機械学習の前処理として，訓練データの正例と負例の数を調整することによってデータの不均衡を解消する．このアプローチは前処理のみに手を

加えることで，既存の機械学習モデルをそのまま適用できるという利点を持っており，不均衡データへの対処法として広く使われている．このアプローチには，正例を増やすオーバーサンプリング (Over sampling) と，負例を減らすアンダーサンプリング (Under sampling) の二種類がある．これらを総称してリサンプリング (Resampling) と呼ぶこともある．オーバーサンプリングの利点としては，訓練データの全サンプルを学習に利用するため，有用なサンプルが捨てられる恐れがないことが挙げらる．一方で，欠点としては，サンプルサイズが大きくなることによる学習の低速化と，正例に対する過学習の恐れがあることが挙げられる．アンダーサンプリングの利点としては，学習の高速化が挙げられ，欠点としては負例の有用なサンプルが捨てられてしまう恐れがあることが挙げられる．このように両者は一長一短であるため，明らかにどちらかが優れているといったことはなく，事例に応じて使い分けたり組み合わせることが望ましい．

2.2.1 オーバーサンプリング

本項では，代表的なオーバーサンプリング手法をいくつか紹介する．

ランダムオーバーサンプリング

正例からランダムに選ばれたサンプルをコピーする．単純に正例のコピーを作るため，後述の SMOTE や ADASYN より高速に動作するが，コピーの対象となった正例に対する過学習が起きやすい．

SMOTE: Synthetic Minority Over-sampling Technique[5]

SMOTE はその名の通り，正例を人工的に生成する手法である．オーバーサンプリングの手法として広く使われており，現在提案されているオーバーサンプリング手法のほとんどがこの手法を拡張したものである．

SMOTE では，まず各正例毎に，その k 個の最近傍の正例から 1 つをランダムに選択する．そして，選ばれた近傍点との線分上のランダムな位置に新たな正例を生成する．この過程を図 2.2 に示す．まず，正例 $\boldsymbol{x}_i$ に対して，k 個 (図では $k = 5$) の最近傍の正例 $\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{i5}$ を探索する．そして，その中からランダムに 1 つ選択する．この例では $\boldsymbol{x}_{i2}$ が選ばれたとする．次に，$\boldsymbol{x}_{i2}$ と $\boldsymbol{x}_i$ の差を変数毎に計算し，それに $[0, 1]$ の乱数を乗じて $\boldsymbol{x}_i$ の値に加えた

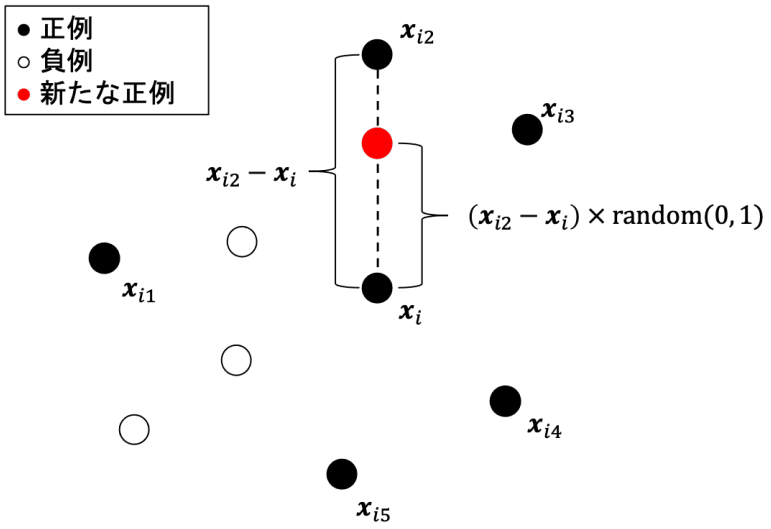

図 2.2: SMOTE による人工サンプル生成過程

ものを新たな正例の値とする．なお，変数が質的変数である場合には，$\boldsymbol{x}_i$ と $\boldsymbol{x}_{i2}$ のどちらかの値を新たな正例の値とする．以上を含む SMOTE の具体的な手順は次の通りである．なお，N^+ は正例の数，$N_{synthetic}$ は新たに生成する正例の数である．また，この手順では，簡単のため，変数が質的変数である場合を考慮していない．

1. $N_{synthetic}$ を N^+ で割った商を q，余りを r とする．$r \neq 0$ ならば，正例を r 個ランダム抽出する．

2. 全ての正例に q 回ずつ，r 個のランダム抽出された正例に 1 回ずつ，以降の処理を適用する．

3. 処理対象の正例を $\boldsymbol{x}_i$ とし，k 個の最近傍の正例 $\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{ik}$ を探索する．

4. $\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{ik}$ から，ランダムに 1 つ選択し，それを $\boldsymbol{x}_{ic}$ とする．

5. 新たな正例 $\boldsymbol{x}_{is} = \boldsymbol{x}_i + R(\boldsymbol{x}_{ic} - \boldsymbol{x}_i)$ を求める．なお，R は $[0, 1]$ の乱数である．

この手法では，内挿により正例を増やすため，正例と負例の分布が入り組んでいない場合に有効であるといえる．逆に，正例同士の間に負例が分布するようなデータセットに対しては，本来取り得ない値を持つ正例を生成してしまう可能性がある．また，$N_{synthetic}$ 回，k 近傍法を実行するため，計算コストが大きい．

ADASYN: Adaptive Synthetic Sampling Approach[6]

ADASYN は SMOTE の拡張手法である．SMOTE では各正例に対して 1 つの新たな正例を生成していたが，ADASYN では各正例の近傍にある負例の数に応じて，生成する新たな正例の数を決定する．その具体的な手順は次の通りである．なお，N はサンプルサイズ，N^+ は正例の数，N^- は負例の数である．

1. 生成する新たな正例の全体数 G を決める．G は $G = (N^- - N^+) \times \beta$ により算出される．ここで $\beta \in (0, 1]$ であり，$\beta = 1$ のとき，ADASYN 適用後のデータセットは完全に均衡となる．

2. 各正例 $\boldsymbol{x}_i (i = 1, ..., N^+)$ について，k 個の最近傍点を探索し，最近傍点における負例の割合 r_i を求める．最近傍点に含まれる負例の数を Δ_i とすると，$r_i = \Delta_i / k$ である．

3. r_i を基準化した値 $\hat{r}_i = r_i / \sum_{i=1}^{N^+} r_i$ を求める．

4. $\boldsymbol{x}_i$ に対して生成する新たな正例の数 g_i を $g_i = \hat{r}_i \times G$ により算出する．

5. $\boldsymbol{x}_i$ の k 個の最近傍点からランダムに g_i 個の正例を選ぶ．

6. SMOTE と同じ要領で $\boldsymbol{x}_i$ と g_i 個の正例の間に新たな正例を生成する．

この手法では，周囲に負例が多い正例の付近に新たな正例が生成されやすい．つまり，クラス境界の決定に寄与する正例が生成されやすい．したがって，正例と負例の分布が入り組んでいないデータセットにおいては，SMOTE より有効に機能する．一方，SMOTE で述べた問題が発生するリスクは SMOTE より高いといえる．

2.2.2 アンダーサンプリング

本項では，代表的なオーバーサンプリング手法をいくつか紹介する．

ランダムアンダーサンプリング

負例からランダムに選ばれたサンプル以外をデータセットから取り除く．オーバーサンプリングではランダムオーバーサンプリングよりも SMOTE が広く使われているが，アンダーサンプリングではこのランダムアンダーサンプリングが広く使われている．ランダム

オーバーサンプリング同様，高速に動作するが，全ての負例を同等に扱うことになるため，有用な負例が取り除かれる可能性がある．

Edited Nearest Neighbours[7]

Edited Nearest Neighbours (ENN) は正例に近い負例を取り除く手法である．クラス境界に近いサンプルを取り除くことで，分類の難易度を下げる．具体的には，各負例について 3 つの最近傍点のうち少なくとも 2 つが正例である場合，その負例を除去する．なお，対象とする最近傍数を 3 より大きくしたり，除去の基準を「最近傍点の中で負例が正例よりも多いこと」や，「最近傍点全てが負例であること」とすることもある．除去する負例を基準に従って決定するため，その数は指定できない．この手法の拡張として，ENN を繰り返し適用する Repeated Edited Nearest Neighbours[8] や，繰り返しの中で徐々に対象とする最近傍数を増加させる AllKNN[8] などがある．

NearMiss[9]

NearMiss では ENN とは対称的に，正例との距離が近い負例を抽出し，それ以外を除去する．NearMiss には 3 つのバージョンが存在し，抽出したい負例の数を n とすると，それぞれの手順は次の通りである．

NearMiss-1 各負例についてそれぞれ 3 つの最近傍の正例との平均距離を計算し，それが最も小さい n 個の負例を抽出する．

NearMiss-2 各負例についてそれぞれ 3 つの最遠方の正例との平均距離を計算し，それが最も小さい n 個の負例を抽出する．

NearMiss-3 各正例に対して，最近傍の $\lceil n/N^+ \rceil$ 個の負例を抽出する．

なお，Zhang らの実験では，これら 3 つの NearMiss に，NearMiss-1 における抽出基準を「平均距離が最も大きい n 個」とした手法 (Distant) とランダムアンダーサンプリングを加えた 5 つが比較されている [9]．指標は F1 値が用いられている．比較の結果，NearMiss-2 とアンダーサンプリングがほぼ同等のスコアであり，NearMiss-1, 3 はアンダーサンプリングよりも悪い結果であったことが報告されている．この手法では，分類が容易であるサンプルを訓練データから取り除き，よりクラス境界の決定に寄与するサンプルのみを残す働きがあるといえる．

表 2.1: コスト行列

	実際は負例	実際は正例
負例と予測	$C(0\|0)$	$C(0\|1)$
正例と予測	$C(1\|0)$	$C(1\|1)$

2.3 アルゴリズムに基づくアプローチ

アルゴリズムに基づくアプローチでは，機械学習のアルゴリズムに手を加えることにより，不均衡データが与える影響を緩和する．そのため，決定木や SVM など，識別器の種類に応じた拡張をする必要があり，汎用性は低い．しかし，データの水増しや削減を行わないため分布が歪められないという利点がある．このアプローチで最も単純な方法は，学習時に正例の重みを大きく，負例の重みを小さくすることである．例えば，正例を誤って負例と予測したときには，負例を誤って正例と予測したときより大きなペナルティを与えることで，より正例の誤分類を抑えられるモデルを構築できる．このような考えに基づく手法として，Cost-Sensitive Learning[10] がある．

Cost-Sensitive Learning

Cost-Sensitive Learning では，サンプル x をクラス i と予測したときのリスク $R(i|x)$ を次のとおり定義し，その最小化を目的とする．

$$R(i|x) = \sum_j P(j|x)C(i|j) \tag{2.3.1}$$

ここで，$P(j|x)$ はサンプル x がクラス j である事後確率であり，$C(i|j)$ はクラス j であるサンプルをクラス i と予測したときのコストである．コストの値はコスト行列によって定義され，二値分類の場合，表 2.1 のようになる．ここでは，クラス 0 が多数クラス，クラス 1 が少数クラスとする．

コスト行列を設定する際には，予測が正解だった場合のコストは予測が外れた場合のそれより低くする必要がある．つまり，$C(1|0) > C(0|0)$ かつ $C(0|1) > C(1|1)$ であり，これを道理性条件 (reasonableness conditions) という．この条件のいずれか一方だけが満たされる場合，片方のクラスだけを予測するだけで常に低いコストを達成できてしまう．例えば $C(1|0) > C(0|0)$ かつ $C(0|1) \leq C(1|1)$ である場合，全て負例と予測することで常にコストの低い予測となる．

表 2.2: 金融業におけるコスト行列の例

	貸し倒れしない人	貸し倒れる人
融資を承認	$-L \times i$	L
融資を拒否	D	0

コスト行列は，適用するタスクに応じて設定する．例えば，金融業において融資を判断するタスクを考える．貸し倒れしない人に融資をした場合は，貸出額に利率を乗じた利息を得る．貸し倒れる人に融資をした場合は，貸出額そのもが損失となる．また，貸し倒れしない人への融資を拒否した場合，顧客からの評判が下がり，収益にマイナスの影響が出る．貸出額を L，利率を i，顧客からの評判が下がることによる減収額を D とすると，コスト行列は表 2.2 のようになる．

Cost-Sensitive Learning を不均衡問題の解決策として利用する場合は，$C(i|i) = 0$ ，$C(j|i) = N^i/N^j$ (N^i はクラス i のサンプルサイズ) とすればよい．これにより，正例を誤って負例と予測したときに大きなコスト，負例を誤って正例と予測したときに小さなコストとなり，正例に対する予測精度の向上が期待できる．

次に，リスクを最小化するモデルの閾値について述べる．まず，サンプル x を正負どちらに予測してもリスクが同じであるとき，

$$R(0|x) = R(1|x) \tag{2.3.2}$$

であり，式 (2.3.1) より，

$$P(0|x)C(0|0) + P(1|x)C(0|1) = P(0|x)C(1|0) + P(1|x)C(1|1) \tag{2.3.3}$$

となる．ここで $p^* = P(1|x)$ とおくと，$P(0|x) + P(1|x) = 1$ より，$1 - p^* = P(0|x)$ となる．したがって，

$$(1 - p^*)C(0|0) + p^*C(0|1) = (1 - p^*)C(1|0) + p^*C(1|1) \tag{2.3.4}$$

となり，これを p^* について整理することで，

$$p^* = \frac{C(1|0) - C(0|0)}{C(1|0) - C(0|0) + C(0|1) - C(1|1)} \tag{2.3.5}$$

が得られる．そして，$P(1|x) \geq p^*$ のときに x を正例と予測するモデルが構築できれば，リスクを最小化することができる．

2.4 アンサンブル学習を活用したアプローチ

データに基づくアプローチでは乱数によって結果が変わる手法が多いため，アンサンブル学習と組み合わせることで，安定して精度の高いモデルを構築できることが期待できる．このアプローチをとった手法は現在までに非常に多く提案されている．そのほとんどがバギング (Bagging) 又はブースティング (Boosting) を活用したものである．

バギングを活用した手法では，訓練データからブートストラップサンプルを作成する際に，リサンプリングを利用することで各ブートストラップサンプルを均衡にする．例えば，SMOTEBagging[11] では，ブートストラップサンプルを作成する際に SMOTE を実行し，均衡なブートストラップサンプルを作成する．また，UnderBagging では，SMOTE の代わりにランダムアンダーサンプリングを利用する．

ブースティングを活用した手法では，各反復でリサンプリングを実行し，得られた均衡な訓練データによって弱識別器を訓練する．例えば，SMOTEBoost[12] では，アダブーストの各反復で SMOTE を実行し，均衡な訓練データを作成する．その均衡な訓練データにより弱識別器を訓練した後，弱識別器の誤差推定値を計算する．SMOTE によって生成した正例は，弱識別器の訓練が終われば除去する．したがって，これらは誤差推定値の計算には使用しない．それ以外の手順はアダブーストと同様である．また，類似の手法で SMOTE の代わりにランダムアンダーサンプリングを利用した RUSBoost (RandomUnderSamplingBoost)[13] も提案されている．なお，RUSBoost などのアンダーサンプリングを利用したアンサンブル手法では，復元抽出によって負例を抽出する．そのため，各ブートストラップサンプルや，各反復で利用される訓練データに含まれる負例は重複する可能性がある．

また，バギングとブースティングを組み合わせた手法として EasyEnsemble[14] がある．この手法では，まずランダムアンダーサンプリングを利用して均衡なブートストラップサンプルを生成し，各ブートストラップサンプルに対してアダブーストを適用する．ブートストラップサンプルの数を T，アダブーストの反復数を s とすると，結果として $T \times s$ 個の弱識別器 $h_{i,j}(i = 1, ..., T; j = 1, ..., s)$ が構築される．予測時にはこの $T \times s$ 個の識別器から得られる出力にアダブーストの各反復で得られた重み $\alpha_{i,j}$ を乗じて総和を取る．この手法では，バギングによって予測の分散を減らし，ブースティングによってバイアスを減らすことが期待できる．さらに，各ブートストラップサンプルに対するアダブーストの適用は並列処理可能なため，バギングとブースティング両者のメリットを有しながらも高速

表 2.3: 評価環境

OS	CentOS release 6.9 (Final)
CPU	Intel Xeon E5-2698 v3 @ 2.30GHz × 2
メモリ	512 GB

な学習が実現できる．EasyEnsemble は，ベルギー税関と同国アントワープ大学が，機械学習を活用した税関不正検知について共同研究を行った際に，不均衡データに対処するために利用され，その有効性がベルギー税関の輸入申告データを用いた実験によって認められている [2].

2.5 各手法の性能評価

本節では，本章で紹介してきた各手法についてその性能を評価する．

2.5.1 性能評価の概要

評価環境を表 2.3 に示す．また，今回利用した機械学習モデル，データセット，性能評価指標は次の通りである．

機械学習モデル

評価対象は各種リサンプリング手法とアンサンブル学習を活用した手法とする．リサンプリングでは，訓練データをリサンプリングした上で決定木 (Desicion Tree; DT) を訓練するモデルを対象とし，ROS (ランダムオーバーサンプリング)+DT ではランダムオーバーサンプリングした上で決定木を訓練する．その他，SMOTE+DT，ADASYN+DT，RUS (ランダムアンダーサンプリング)+DT，RENN (Repeated Edited Nearest Neighbours)+DT，NM2 (Neamiss-2)+DT を対象とする．アンサンブル学習を活用した手法としては，ベースラインとしてバギング，アダブースト，不均衡に対応した手法として SMOTEBagging, SMOTEBoost, UnderBagging, RUSBoost, EasyEnsemble を対象とする．アンサンブル学習における弱識別器は全て決定木とする．弱識別器の数について，Fernández らによって行われた実験を参考に, バギングは 40, ブースティングは 10 とする [15]．なお, EasyEnsemble については 10 個のブートストラップサンプルそれぞれに対し，反復数 4 のブースティン

グを行うことで，最終的な弱識別器の数を 40 とする．また，本評価で利用するリサンプリング等の実装は Python の imbalanced-learn[16] パッケージを利用し，決定木の実装は scikit-learn パッケージを利用する．

データセット

imbalanced-learn パッケージに含まれる 27 個のデータセットを利用する．これらのデータセットは Ding によってまとめられたもので，様々な不均衡比率，サンプルサイズ，特徴量数，特徴量の種類及びドメインを網羅するように考慮されている [17]．利用するデータセットの概要を表 2.4 に示す．データセットの中には，目的変数が多値のものや連続変数のものも含まれている．そのため，「少数クラス」で定義される値を少数クラス，それ以外を多数クラスとして，二値分類用に目的変数の値が変更されている．例えば，optical_digits は手書き数字の認識問題であり，0 から 9 までの 10 クラス存在するため，8 を少数クラス，それ以外を多数クラスとすることで不均衡な二値分類用のデータセットとしてる．「特徴量の種類」の N は質的変数 (nominal)，C は量的変数 (continuous)，B は二値変数 (binary) を意味し，それぞれの数を表している．「特徴量数」は質的変数をダミー変数化した最終的な変数の数である．

性能評価指標

性能評価は，訓練に要した秒数 (Time[s])，AUC-ROC，AUC-PR，正解率 (Accuracy)，再現率 (Recall)，適合率 (Precision)，F1 値，マシューズ相関係数 (Matthews Correlation Coefficient; MCC)，G-mean の 9 つにより行う．AUC-ROC は，2.1 節で紹介した，ROC 曲線の下側面積である．後述の AUC-PR と区別するために，以降は AUC-ROC と表記する．AUC-PR は，Precision-Recall 曲線の下側面積である．Precision-Recall 曲線は，ROC 曲線と同様に閾値を 0 から 1 まで変化させ，適合率を縦軸，再現率を横軸としたグラフにそれぞれの値をプロットしたものである．AUC-ROC では正例と負例双方の予測の正しさが評価されるため，不均衡なデータでは正例の予測が悪くても比較的高いスコアが出る．一方，AUC-PR は正例の予測に焦点を当てた指標であるため，不均衡なデータにおけるモデルの性能差をより明確に捉えることが期待できる．Saito らは不均衡データ分類タスクにおいて，ROC 曲線よりも PR 曲線によってモデルの性能を可視化することを推奨している

表 2.4: データセットの概要

ID	データセット名	リポジトリ	少数クラス	不均衡比率	サンプルサイズ	特徴量数	特徴量の種類	ドメイン
1	ecoli	UCI	imU	8.6	336	7	7C	Life
2	optical_digits	UCI	8	9.1	5,620	64	64C	Computer
3	satimage	UCI	4	9.3	6,435	36	36C	Physical
4	pen_digits	UCI	5	9.4	10,992	16	16C	Computer
5	abalone	UCI	7	9.7	4,177	10	7C, 1N	Life
6	sick_euthyroid	UCI	sick euthyroid	9.8	3,163	42	7C, 18N	Life
7	spectrometer	UCI	>= 44	11	531	93	93C	Physical
8	car_eval_34	UCI	good, v good	12	1,728	21	6N	Business
9	isolet	UCI	A, B	12	7,797	617	617C	Computer
10	us_crime	UCI	> 0.65	12	1,994	100	100C	Social
11	yeast_ml8	LIBSVM	8	13	2,417	103	103C	Life
12	scene	LIBSVM	> *onelabel*	13	2,407	294	294C	Nature
13	libras_move	UCI	1	14	360	90	90C	Physical
14	thyroid_sick	UCI	sick	15	3,772	52	7C, 21N	Life
15	coil_2000	KDD, CoIL	minority	16	9,822	85	85C	Social
16	arrhythmia	UCI	06	17	452	278	206C, 73N	Biology
17	solar_flare_m0	UCI	$M->0$	19	1,389	32	10N	Nature
18	oil	UCI	minority	22	937	49	49C	Environment
19	car_eval_4	UCI	vgood	26	1,728	21	6N	Business
20	wine_quality	UCI, wine	<= 4	26	4,898	11	11C	Business
21	letter_img	UCI	Z	26	20,000	16	16C	Computer
22	yeast_me2	UCI	ME2	28	1,484	8	8C	Life
23	webpage	LIBSVM, w7a	minority	33	34,780	300	300B	Web
24	ozone_level	UCI	ozone, data	34	2,536	72	72C	Environment
25	mammography	UCI	minority	42	11,183	6	6C	Life
26	protein_homo	KDD CUP 2004	minority	111	145,751	74	74C	Biology
27	abalone_19	UCI	19	130	4,177	10	7C, 1N	Life

[18]．なお，AUC-PR は AUC-ROC と異なり 0 から 1 までの値をとる．マシューズ相関係数は，不均衡データを扱うことの多い生物情報学の分野を起源とする評価指標である．正例の数，負例の数をそれぞれ N_{P}，N_{N}，正例と予測した数 (TP+FP)，負例と予測した数 (TN+FN) をそれぞれ P_{P}，P_{N} とすると，次式で与えられる．

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} + \mathrm{FP} \times \mathrm{FN}}{\sqrt{N_{\mathrm{P}} \times N_{\mathrm{N}} \times P_{\mathrm{P}} \times P_{\mathrm{N}}}} \tag{2.5.1}$$

この指標の特徴として，混同行列の全ての値が良いときに高い値となる．また，予測が全て誤りであったときに-1，ランダムのときに 0，全て正解であったときに 1 をとる．予測が全て誤りであったときは，その予測を反転させれば全て正解となるため，実質的には 0 が最も悪く，1 が最も良いということになる．

表 2.5: imbalanced-learn パッケージのデータセットを用いた評価結果

model	Time [s]	AUCROC	AUCPR	Accuracy	Recall	Precision	F1	MCC	Gmean
ROS+DT	1.06	0.7478	0.5409	0.9447	0.5312	0.5410	0.5285	0.5033	0.6777
SMOTE+DT	2.66	0.7681	0.5399	0.9334	0.5777	0.4848	0.5153	0.4889	0.7121
ADASYN+DT	2.92	0.7634	0.5325	0.9320	0.5689	0.4764	0.5066	0.4794	0.7045
RUS+DT	0.02	0.8129	0.5575	0.8081	0.8176	0.2879	0.3930	0.3957	0.8105
RENN+DT	0.74	0.7672	0.5559	0.9443	0.5636	0.5307	0.5395	0.5141	0.6971
NM2+DT	0.02	0.6503	0.5096	0.4496	0.8738	0.1367	0.2134	0.1707	0.5694
Bagging	5.74	0.8872	0.6217	0.9637	0.5023	0.6754	0.5528	0.5535	0.6299
AdaBoost	0.65	0.8823	0.5233	0.9564	0.4247	0.5833	0.4671	0.4653	0.5637
SMOTEBagging	27.57	0.9028	0.6153	0.9613	0.5385	0.6322	0.5685	0.5582	0.6649
SMOTEBoost	279.88	0.8806	0.5828	0.9439	0.6170	0.5423	0.5583	0.5407	0.7368
UnderBagging	0.86	0.9138	0.6123	0.8991	0.8113	0.4288	0.5269	0.5288	0.8509
RUSBoost	0.14	0.8655	0.4681	0.8460	0.7626	0.3086	0.4111	0.4087	0.7969
EasyEnsemble	0.30	0.8916	0.5117	0.8276	0.8405	0.2787	0.3966	0.4084	0.8300

2.5.2 評価結果

評価結果を表 2.5 に示す．この評価では，5 分割の交差検証を用い，その分割はランダムシードを変えて 5 回行っている．さらに，各モデルはランダムシードを変えて 5 回訓練している．よって，データセットとモデルの組み合わせ毎に合計で 125 回の訓練が行われたことになる．表中の数値は各データセット毎に 125 回の平均値をとり，それを全データセットについて平均したものである．また，各指標毎に，値が大きいほど濃い赤色に着色している．

オーバーサンプリング手法の比較

オーバーサンプリング後に決定木を訓練したモデルについては，大きな違いはみられなかった．しかし，SMOTE より ADASYN が，全指標においてわずかに悪い結果となった．これは，2.2.1 項で述べた問題点が発生しているためと考えられる．

アンダーサンプリング手法の比較

アンダーサンプリング後に決定木を訓練したモデルについては, NearMiss のスコアの低さが目立つ．今回評価したモデルの中でも最も悪い結果となっている．ADASYN が SMOTE より悪かったことも考慮すると，クラス境界に近いデータを活用することが精度を下げる要因となったと考えられる．一方，Zhang らの実験では，NearMiss-2 がランダムアンダー

サンプリングとほぼ同じ F1 値となっている [9]．これは，評価に使用されたデータセットが NearMiss-2 に有利なものであったためと考えられる．また，アンダーサンプリングと Repeated Edited Nearest Neighbours の優劣については，この結果からは判断し難い．

オーバーサンプリングとアンダーサンプリングの比較

おおよその傾向として，オーバーサンプリングを利用した手法では，正解率，適合率，F1 値，マシューズ相関係数が高く，再現率や G-mean が低い．一方，アンダーサンプリングを利用した手法では，その逆の傾向がみられる．これらのうち，適合率と再現率については，一般的にトレードオフの関係にあるため，当然の結果といえる．適合率を上げるための簡単な方法は，正例と予測する数を減らすことである．しかし，正例と予測する数が少なくなれば，正例に対する正解率，つまり再現率は下がる．G-mean については，TPR = Recall であって，TNR は負例の多い不均衡データにおいて変化を受けにくいという理由から，再現率とほぼ同じように変化していると考えられる．また，適合率が高く再現率が低いモデルは，先述の理由から，正例と予測する数が少ないモデルである可能性が高い．不均衡データにおいては，そのようなモデルは正解率が高い．以上より，オーバーサンプリングとアンダーサンプリングの性能について，これらの指標により優劣を付けることは困難である．ただし，訓練に要する時間については，オーバーサンプリングの方が長い．特に，アンサンブル学習と組み合わせると，その傾向は顕著に現れ，並列処理のできない SMOTEBoost では訓練に非常に長い時間を要している．これは，より大きなサンプルサイズのデータセットを扱う税関不正検知への適用を考えると，大きな欠点である．

単体の決定木とアンサンブル学習の比較

SMOTE+DT と比べ，SMOTEBagging，SMOTEBoost は AUC-ROC，AUC-PR，マシューズ相関係数が高い．特に，SMOTEBoost については，訓練時間を除く全ての指標において SMOTE+DT より良い結果となった．また，RUS+DT に比べ，RUSBoost，EasyEnsemble は AUC-ROC が高いが，他の指標では有意に優れているとは言い難い．一方，UnderBagging については，訓練時間と再現率を除く全ての指標において RUS+DT より良い結果となっており，なおかつ再現率の差は約 0.006 とごく僅かである．以上から，アンサンブル学習の手法によっては，単体の決定木より良いモデルを構築できるといえる．

第3章　税関不正検知への応用

第 2 章では，imbalanced-laern パッケージに含まれるデータセットを用いて既存手法の性能を評価した．本章ではそれらを税関不正検知に対して適用し，その有効性を検証する．

この検証では，税関不正検知の対象を輸入申告における不正とする．輸入申告とは，輸入者が外国貨物を国内に輸入するために必要な手続き (輸入通関) において，輸入者から税関へ，その外国貨物を輸入する旨を申告することである．税関はその輸入申告について，必要な審査・検査を実施する．その結果，問題がなく，また輸入に必要な関税等が納付されたことが確認されれば，輸入を許可する．輸入申告の際には，税関へ輸入申告書が提出される．輸入申告書に必要な記載事項は法令や政令で定められている．日本の場合は，関税法施行令第 59 条の各号に掲げられている通り，貨物の品名，数量，価格や，仕出人の名称，居所などを記載する必要がある．税関手続きは全世界で電子化が進められており．現在日本ではほとんど全ての輸入申告が電子的に行われている．電子的に行われた輸入申告のデータを蓄積し，それが不正であったか否かを示すラベルを申告単位で付与したものを，この検証で扱うデータセットとする．輸入申告のデータは，Kim らによって人工的に作成された疑似データ[1]と，税関に蓄積された実データの 2 種類を使用し，それぞれで検証を行う．

3.1　疑似データを用いた検証

疑似データについては，敵対的生成ネットワーク (GAN) の技術により生成されており，インターネット上で公開されているものに対して分析を行った．サンプルサイズは 100,000 で，そのうち 6,850 件が不正とラベル付けされている．特徴量は表 3.1 に示す通り 11 項目ある．sgd.id は各申告の ID，つまり，各サンプルの ID であり，特徴量とはならないため，その数に計上していない．

輸入申告には通関に関する専門知識が必要であることから，輸入者の代理人が輸入申告を行うことがある．そのため，輸入者 (impoerter) と申告者 (declarant) それぞれに ID を

[1] https://github.com/Roytsai27/Dual-Attentive-Tree-aware-Embedding

表 3.1: 疑似データの特徴量

項目名	内容	例
sgd.id	輸入申告の ID	SGD2
sgd.date	輸入申告の日付	13-01-02
importer.id	輸入者の ID	IMP837219
declarant.id	申告者の ID	DEC1525
country	貨物の仕出国	CNTRY680
office.id	申告を処理した部署の ID	OFFICE51
tariff.code	貨物の品目に応じて定義されるコード	8703232926
quantity	貨物の数量	1
gross.weight	貨物の重さ	3,910kg
fob.value	貨物の FOB 価格	$39,000
cif.value	貨物の CIF 価格	$40,000
total.taxes	税額	$500

付与しており，importer.id と declarant.id は別の意味を持つ項目となっている．ただし，これらは後述の通り，ユニークな値の数（重複を除いた値の数）が多く今回の検証では使用していない．tariff.code は輸入貨物の品目に応じて定義され，先頭の 6 桁は，通称「HS 条約」と呼ばれる「商品の名称及び分類についての統一システムに関する国際条約」で定められているコード (HS コード) である．この 6 桁のうち，先頭 2 桁を類 (Chapter) 、類を含む先頭 4 桁を項 (Heading) 、項を含む先頭 6 桁を号 (Sub-heading) と呼ぶ．表に示した「8703232926」を例にとると，この貨物は第 87 類「鉄道用及び軌道用以外の車両並びにその部分品及び附属品」の中の第 8703 項「乗用自動車、自動車、ステーションワゴン、レーシングカー」である．なお，HS 番号を除いた 7 桁目以降の番号は国によって異なる．

fob.value，cif.value における FOB (Free On Board) 及び CIF (Cost, Insurance and Freight) は，インコタームズという国際規則で定められた貿易条件で，貿易にかかる費用を売主と買主がどこまで負担するかについて定めたものである．FOB と CIF の違いは，輸送料及び保険料の負担主の違いである．前者であれば買主が負担し，後者であれば売主が負担する．したがって，FOB 価格と CIF 価格の差額は輸送料及び保険料となる．

質的変数のユニークな値の数について確認したところ，importer.id が 8,653，declarant.id

表 3.2: 疑似データを用いた検証結果

model	Time [s]	AUCROC	AUCPR	Accuracy	Recall	Precision	F1	MCC	Gmean
ROS+DT	12.47	0.5895	0.2597	0.8731	0.2569	0.2078	0.2297	0.1625	0.4864
SMOTE+DT	8.10	0.6065	0.2824	0.8311	0.3431	0.1734	0.2303	0.1576	0.5462
ADASYN+DT	6.69	0.6030	0.2781	0.8285	0.3385	0.1689	0.2253	0.1517	0.5418
RUS+DT	0.50	0.6695	0.4168	0.6682	0.6709	0.1385	0.2296	0.1848	0.6694
RENN+DT	7.04	0.5750	0.2380	0.8754	0.2227	0.1959	0.2084	0.1415	0.4544
NM2+DT	1.04	0.5065	0.4848	0.1824	0.8866	0.0747	0.1378	0.0102	0.3347
Bagging	14.35	0.7813	0.2166	0.9239	0.0227	0.2888	0.0421	0.0628	0.1495
AdaBoost	2.71	0.7994	0.2856	0.9263	0.0000	0.0000	0.0000	0.0000	0.0000
SMOTEBagging	285.24	0.7862	0.2297	0.9059	0.1953	0.2925	0.2342	0.1905	0.4334
UnderBagging	4.48	0.8105	0.2408	0.8030	0.6731	0.2229	0.3349	0.3056	0.7399
RUSBoost	1.82	0.8000	0.2706	0.7272	0.7693	0.1822	0.2942	0.2787	0.7457
EasyEnsemble	1.34	0.7835	0.4341	0.7436	0.7064	0.1815	0.2888	0.2617	0.7262

が 1,468 と非常に多いため，これらの特徴量を削除した．tariff.code についても 1,894 と多いが，先頭 2 桁 (類) のみであればそのユニーク数は 100 以下であるため，それを新たな特徴量として使用した．輸入申告の日付である sgd.date については，曜日を表すフラグと，月の上旬 (1 日〜10 日) , 中旬 (11 日〜20 日) , 下旬 (21 日〜月末) を表すフラグへと変換し，使用した．なお，曜日毎の輸入申告件数を確認したところ，平日が 20,000 件前後であるのに対し，土曜日は 373 件，日曜日は 76 件であり，直感的に理解できる傾向を持っていることが確認できた．量的変数については Kim らの実験 [1] と同様に，unit.value (cif.value/quantity)，weight.value (cif.value/gross.weight)，tax.ratio (total.taxes/cif.value)，unit.tax (total.taxes/quantity)，face.ratio (fob.value/cif.value) を新たな特徴量として作成し，元々の量的変数に加えてこれらも使用した．評価環境は 2.5 項と同様，表 2.3 の通りである．

検証結果を表 3.2 に示す．この検証では，sgd.date が 2013 年 1 月〜11 月のものを訓練データ，12 月のものテストデータとしている．訓練データのサンプルサイズは 90,107，うち正例が 6,850，不均衡比率は約 12.15 である．テストデータのサンプルサイズは 9,893，うち正例が 729，不均衡比率は約 12.57 である．特徴量は，ダミー変数化等の前処理によって 241 項目となっており，うち 231 項目がダミー変数である．また，表の値は，各モデルのランダムシードを変えてそれぞれ 20 回訓練し，その平均値をとったものである．対象とするモデルは，2.5 節と同じである．ただし，SMOTEBoost は数日間実行しても計算が終了しなかったことから，評価できなかった．

今回の結果では，2.5 節での結果と比べて，単体の決定木とアンサンブル手法の差がより明確となっている．特に，UnderBagging，RUSBoost，EasyEnsemble については，F1

表 3.3: 実データを用いた検証結果

model	Time [s]	AUCROC	AUCPR	Accuracy	Recall	Precision	F1	MCC	Gmean
Bagging	18.93	0.5555	0.0643	0.9994	0.0179	0.4500	0.0343	0.0889	0.0945
AdaBoost	144.63	0.8581	0.0070	0.9994	0.0000	0.0000	0.0000	0.0000	0.0000
UnderBagging	8.05	0.8919	0.0813	0.8540	0.7214	0.0028	0.0056	0.0389	0.7843
RUSBoost	29.77	0.8712	0.0044	0.8411	0.6661	0.0024	0.0047	0.0330	0.7451
EasyEnsemble	7.92	0.8871	0.0043	0.7849	0.8375	0.0022	0.0044	0.0361	0.8102
Deep Neural Network	-	0.8379	0.0018	0.8431	0.7143	0.0026	0.0051	0.0365	0.7761

値，マシューズ相関係数，G-mean において，その他の手法より明らかに優れている．また，これらは再現率も比較的高い．つまり，正例を誤って負例と予測する可能性が比較的低い．以上から，アンダーサンプリングとアンサンブル学習の組み合わせが，税関不正検知に有効であるといえる．なお，リサンプリングしていない 2 つのモデル（バギング，アダブースト）については，ほとんどのスコアで他より悪い結果となっている．特に，アダブーストは再現率も適合率も 0 であることから，全て負例と予測するモデルが構築されている．今回，2.5 節の結果と比べてこのような差が出た原因としては，特徴量のほとんどがダミー変数であることが考えられる．

3.2 実データを用いた検証

税関の実データについては，機密性確保のため詳細は開示できないが，ある連続期間の約 656,600 件の輸入申告データであり，うち約 100 件が不正のあった申告としてラベル付けされている．特徴量は，ダミー変数化等の前処理を行っていない状態で 33 項目である．このデータセットに含まれる輸入申告データは，全て税関職員による検査が行われた申告のデータであり，検査時の見逃しがない限りは正しくラベル付けされているといえる．

前節の結果から，アンダーサンプリングとアンサンブル学習の組み合わせが，税関不正検知において有効であることが確認できた．そのため，それらの手法とベースラインであるバギングとアダブーストを実データに適用し効果を検証する．また，この検証では，深層学習 (Deep Neural Network) との比較も行う．深層学習の訓練データは，ランダムオーバーサンプリングにより均衡なデータセットとしている．深層学習のネットワーク構成及び評価環境についても，機密性確保のため開示できない．

検証結果を表 3.3 に示す．深層学習については，他のモデルと異なる環境で実行しているため，訓練に要した時間を計測していない．また，複数パターンの構成で試行し，最も

良かった構成での結果を記載している．

今回の結果からも，アンダーサンプリングとアンサンブル学習の組み合わせが有効であることが確認できる．それらのスコアが深層学習と同等か，場合によっては高いものとなっている．深層学習モデルのチューニングや訓練コストを考慮すると，この結果は本研究の一つの貢献といえる．なお，バギングについては，F1 値やマシューズ相関係数が非常に高いものの，再現率や AUC-ROC が非常に低い．もし，F1 値やマシューズ相関係数のみによりモデルの優劣を判断すれば，望まない結果となってしまう．このように，複数の指標によりモデルを比較することは重要であるといえる．

第4章　既存手法の改善策の検討及び評価

前章の結果から，アンダーサンプリングとアンサンブル学習の組み合わせが，税関不正検知に対して効果的であることが確認できた．そのため，本章ではその改善策について検討し，評価を行う．

4.1　CUS: Clustering Based Undersampling

アンダーサンプリングの改善策の一つに，クラスタリングベースのアンダーサンプリング (Clustering Based Undersampling; CUS) がある．クラスタリングを負例のみに適用することもあれば，全サンプルに適用することもあり，また，クラスタを基にどのようにアンダーサンプリングを行うかは手法によって様々である．以降，クラスタリングを活用したアンダーサンプリング手法全般を CUS と呼ぶこととする．

4.1.1　CUS とアンサンブル学習の組み合わせ

CUS とアンサンブル学習を組み合わせた手法も提案されている．ここでは，そのような手法のうち，ClusterBal[19] と CUSBoost[20] について紹介する．

ClusterBal

ClusterBal は，アンダーサンプリングで有用な情報が捨てられてしまうという欠点を補うために提案された手法である．この手法で目指したことは，全ての負例が必ずいずれかの弱識別器の訓練データに含まれるようにし，なおかつ各弱識別器の訓練データに含まれる負例が全て異なるようにすることである．そのために，まずは全ての負例を，正例と同数のサンプルを持つ集合に分割する．全ての負例の数を N^-，i 番目 $(i = 1, ..., S)$ の集合のサンプルサイズを N_i^-，全ての正例の数を N^+ とすると，$N_i^- \fallingdotseq N^+$ となる．S は N^- を N^+ で割った商とする．N_i^- と N^+ が「おおよそ等しい」となるのは，N^- が N^+ で割り切れ

るとは限らないからである．i 番目の弱識別器の訓練データは，i 番目の集合に属する負例 (N_i^- 個) と，全ての正例 (N^+ 個) で構成される．$N_i^- \fallingdotseq N^+$ であるため，各弱識別器の訓練データはほぼ均衡となる．負例の各集合はクラスタリングによって得る．k-means($k = S$) を負例に適用し，得られた各クラスタを負例の各集合とする．k-means では各クラスタが同じサンプルサイズになることを保証しないが，Sun らによれば，負例同士は何かしらの共通点を共有しており，あるクラスタに属する負例を他のクラスタに入れて調整を行っても問題ない [19]．こうして作成された均衡な各弱識別器の訓練データに対し，それぞれ弱識別器を訓練する．予測時には，各弱識別器の出力値を独自の集計ルール (ensemble rule) によって集計し，モデルの最終的な出力とする．集計ルールについては 5 種類が提案されているが，それらの中でも MaxDistance というルールが最も良い結果であったことが報告されている．全ての集計ルールの共通事項は，予測対象のサンプルについて，多数クラスと少数クラスに対するスコアをそれぞれ計算し，スコアが高い方をそのモデルの予測値とすることである．そして，各ルールの違いは，そのスコアの算出方法にある．MaxDistance における，多数クラスに対するスコア R^- は次式によって算出される．

$$R^- = \max_{1 \leq i \leq S} \frac{P_i^-}{D_i^- + 1} \tag{4.1.1}$$

ここで，P_i^- は i 番目の弱識別器による，多数クラスに対する予測確率で，D_i^- は予測対象のサンプルと i 番目の集合に属する各負例との距離の平均値である．少数クラスに対するスコア R^+ についても同様に算出される．Sun らによる検証では，AUC-ROC によって，ClusterBal と MaxDistance の組み合わせ (ClusterBal+MaxDistance) を，SMOTEBoost，RUSBoost，UnderBagging，EasyEnsemble 等と比較している [19]．その結果，提案手法が最も良い性能であったことが報告されている．

CUSBoost

CUSBoost は，RUSBoost における各反復で利用するランダムアンダーサンプリングを，CUS に置き換えたものである．この手法における CUS では，まず訓練データの負例のみに対して k-means を適用する．そして，得られた各クラスタからそれぞれ 50%の負例を抽出する．k-means の k はハイパーパラメータであり，各クラスタから抽出する割合も必要に応じて変更可能である．Rayhan らは，AUC-ROC によって，この手法を AdaBoost，

RUSBoost, SMOTEBoost と比較している [20]．その結果, 提案手法が最も良い性能であったことが報告されている．

4.1.2 既存手法の評価

Sun らによる検証では，ClusterBal と MaxDistance の組み合わせを評価しており，評価結果が ClusterBal によるものかどうかは議論されていない [19]．また，Sun らによる検証と Rayhan らによる検証では AUC-ROC のみで評価している [19][20] が，この指標は先述の通り，正例に対する予測精度が悪くても高いスコアが出ることがあるため，他の指標でも評価する必要があると考える．したがって，ClusterBal，ClusterBal+MaxDistance，CUSBoost について，これまで利用してきた指標により評価を行う．

ClusterBal 及び MaxDistance については，Sun らの論文 [19] を参考に実装した．なお，先述の通り，k-means は各クラスタのサンプルサイズがほぼ同数になることを保証しないため，制約付き k-means (Constrained K-Means)[21] の Python による実装[1]を利用した．この手法によって，各クラスタのサンプルサイズに下限や上限を設けることができる．今回の実装にあたり，各クラスタのサンプルサイズの下限を N^+ とした，MaxDistance を利用しない ClusterBal では，モデルの予測確率を弱識別の予測確率の平均値とし，予測確率が高い方のクラスを，そのモデルの予測値とした．これは，scikit-learn パッケージのバギング (BaggingClassifier) と同様である．ClusterBal+MaxDistance では，式 (4.1.1) により算出される R^- を予測確率 P^- に変換するために，$P^- = \frac{R^-}{R^- + R^+}$ とした．R^+ についても同様である．なお，弱識別器は EasyEnsemble 同様，反復数 4 のアダブーストとした．

CUSBoost については，Rayhan らによって公開されている Python の実装[2]を参考に実装した．Rayhan らによる実装をそのまま利用しなかった理由は，いくつかのバグが存在していたからである．ハイパーパラメータである k-means の k は，Rayhan らによる実装に合わせて 23 とした．

評価に使用するデータセットは，scikit-learn パッケージの make_classification 関数により人工的に生成した．その特徴量は 20 項目，サンプルサイズは 5,000 で，うち正例の数は 100 である．

評価結果を表 4.1 に示す．この評価では，5 分割の交差検証を用い，その分割はランダム

[1] https://github.com/joshlk/k-means-constrained
[2] https://github.com/farshidrayhan-uom/CUSBoost

表 4.1: CUS を活用したアンサンブル手法の評価結果

model	AUCROC	AUCPR	Accuracy	Recall	Precision	F1	MCC	Gmean
Bagging	0.8248	0.2816	0.9809	0.0940	0.6985	0.1605	0.2421	0.2928
AdaBoost	0.7562	0.0872	0.9773	0.0100	0.0507	0.0166	0.0159	0.0394
SMOTEBagging	0.9203	0.4188	0.9829	0.2320	0.7393	0.3483	0.4043	0.4757
SMOTEBoost	0.7568	0.1849	0.9283	0.4460	0.1294	0.1998	0.2120	0.6427
UnderBagging	0.8656	0.2783	0.8808	0.6780	0.1081	0.1864	0.2380	0.7716
RUSBoost	0.7356	0.0899	0.7403	0.6140	0.0491	0.0906	0.1169	0.6671
EasyEnsemble	0.7966	0.1069	0.7734	0.6640	0.0574	0.1055	0.1461	0.7143
ClusterBal	0.7651	0.1071	0.0200	1.0000	0.0200	0.0392	0.0000	0.0000
ClusterBal+MaxDistance	0.8908	0.2764	0.9472	0.5220	0.2003	0.2879	0.3005	0.7047
CUSBoost	0.7642	0.0869	0.9740	0.0500	0.1127	0.0667	0.0621	0.1558

シードを変えて 5 回行った．つまり，モデル毎に 25 回の訓練が行われており，表に記載の値はその平均値である．

Sun らによって報告されている通り，ClusterBal+MaxDistance は，SMOTEBoost，UnderBagging，RUSBoost，EasyEnsemble より高い AUC-ROC となっている．また，他の指標でも比較的良い結果となっている．しかし，比較対象としたモデルより必ずしも良いとは結論づけ難い．CUSBoost についても，AdaBoost や RUSBoost より高い AUC-ROC であり，Rayhan らの報告内容と一致する．しかし，CUSBoost については再現率，F1 値，マシューズ相関係数，G-mean が著しく低い．ClusterBal については，マシューズ相関係数，G-mean が 0 となってしまっている．正解率などの指標からも，ClusterBal ではほぼ全てを正例と予測し，CUSBoost ではほぼ全てを負例と予測していると考えられる．

ClusterBal 及び CUSBoost が，このような結果となった原因を明らかにするため，それぞれの弱識別器の決定境界を可視化した．その結果を図 4.1 及び 4.2 に示す．このグラフは，主成分分析 (PCA) により特徴量を 2 次元に圧縮している．また，それぞれ先頭から 4 つの弱識別器 (baseclassifier) の決定境界を可視化している．黄色に着色されている領域が，その弱識別器が正例と予測する領域である．凡例の「Neg」，「Pos」はそれぞれ負例，正例で，「Neg[0]」は 0 番目の弱識別器の訓練データに属する負例である．つまり，0 番目の弱識別器は「Pos」と「Neg[0]」で訓練されている．

ClusterBal では，正例と予測しがちな弱識別器が構築されていることが分かる．各弱識別器の訓練データに着目すると，正例は空間内で広く散らばっているが，負例は狭い範囲に分布している．これは，負例がクラスタそのものだからである．負例の分布する範囲に比べ，正例のそれが非常に広いことが，このような決定境界を生み出す要因となっている

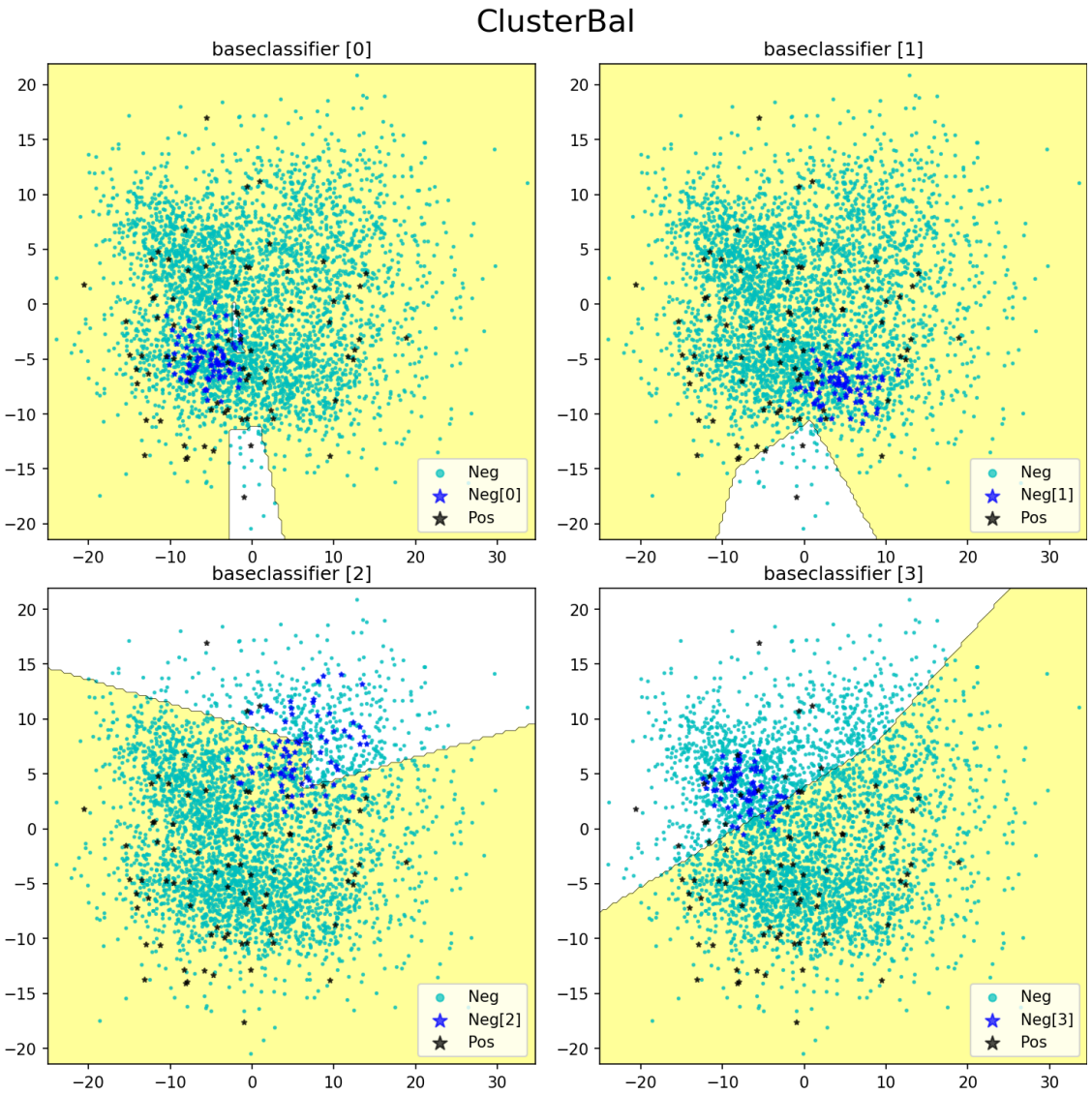

図 4.1: ClusterBal の弱識別器の決定境界

と考えられる．

一方，CUSBoost では，負例と予測しがちな弱識別器が構築されていることが分かる．そして，弱識別器の訓練データが依然として不均衡であることが分かる．これは，負例の各クラスタから 50%ずつ抽出しているためである．今回使用したデータセットは負例の数が 4900，正例の数が 100 であり，不均衡比率は 49 である．この負例を 50%抽出して弱識別器の訓練データを作成するため，各弱識別器の訓練データにおける不均衡比率は $4900 \times 0.5/100 = 24.5$ となる．そのため，このように負例の領域が広い決定境界となっていると考えられる．

以上から，ClusterBal では負例の各クラスタを抽出し，それぞれを各弱識別器の訓練デー

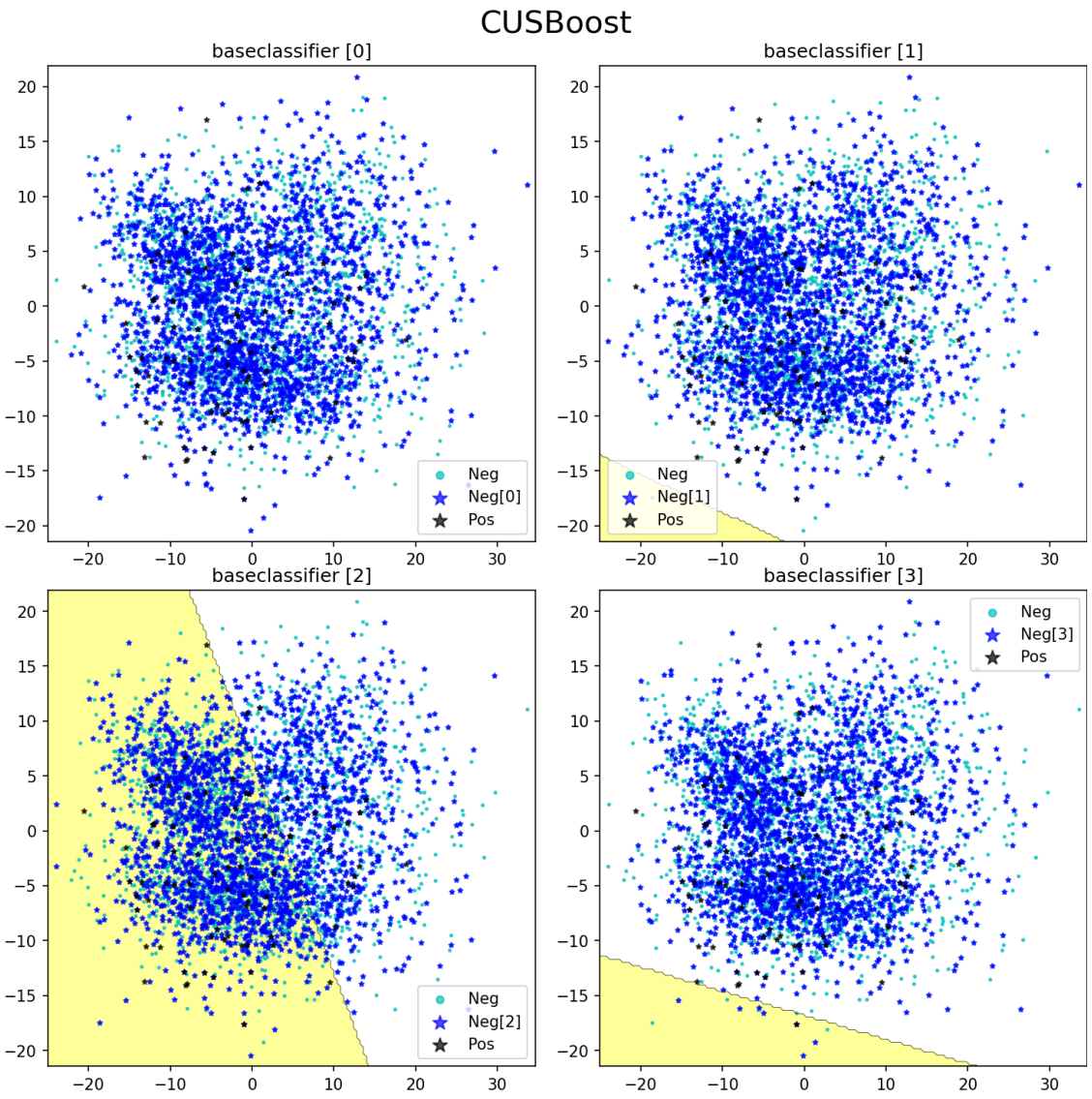

図 4.2: CUSBoost の弱識別器の決定境界

タとすることに問題があり，CUSBoost では負例の各クラスタから 50%抽出することで，依然として不均衡が解消されていないことに問題があるといえる．

4.2 CUS の改善

これらの問題を解決するため，負例の各クラスタから抽出した負例の合計数を正例の数と等しくする方法を提案する．つまり，i 番目のクラスタ c_i のサンプルサイズを N_i^- とすると，c_i から抽出する負例の数 S_i は，次式によって算出すればよい．

$$S_i = N^+ \times \frac{N_i^-}{N^-} \tag{4.2.1}$$

表 4.2: 提案手法の評価結果

model	AUCROC	AUCPR	Accuracy	Recall	Precision	F1	MCC	Gmean
UnderBagging	0.8663	0.2831	0.8774	0.6860	0.1072	0.1851	0.2378	0.7754
CUSBagging	0.8652	0.2839	0.8626	0.7060	0.0984	0.1724	0.2284	0.7800
UMAP+CUSBagging	0.8679	0.2870	0.8611	0.7140	0.0975	0.1714	0.2290	0.7837
RUSBoost	0.7280	0.0683	0.7577	0.5800	0.0484	0.0890	0.1125	0.6604
CUSBoost_TT	0.7130	0.0678	0.7003	0.6260	0.0446	0.0825	0.1050	0.6496
UMAP+CUSBoost_TT	0.7019	0.0703	0.7237	0.5740	0.0437	0.0807	0.0976	0.6332
EasyEnsemble	0.7971	0.0951	0.7696	0.7020	0.0592	0.1090	0.1558	0.7333
CUSEasyEnsemble	0.8010	0.1049	0.7344	0.7160	0.0522	0.0973	0.1415	0.7221
UMAP+CUSEasyEnsemble	0.7982	0.0990	0.7424	0.7060	0.0537	0.0997	0.1431	0.7218

$\sum_{i=1}^{k} S_i = N^+$ であるため，各弱識別器の訓練データは均衡となる．c_i から S_i 個のサンプルの抽出はランダム抽出とする．この手法では，負例の各クラスタから負例を抽出するため，各弱識別器の訓練データの負例が広い範囲に分布することが期待できる．

また，既存の CUS では，訓練データに直接 k-means を適用しているものがほとんどである．そこで，k-means の適用前に，UMAP[22] によって次元圧縮することを提案する．UMAP によって，局所的・大域的な特徴を捉えて次元圧縮された特徴量空間においてクラスタリングを行うことで，よりまとまりのあるクラスタを得られることが期待できる．

次に，これらの手法を実装し，前節と同じく人工的に生成したデータセットにより評価を行う．今回は，UnderBagging，RUSBoost，EasyEnsemble で利用されるランダムアンダーサンプリングを，提案するアンダーサンプリング手法に置き換えた．

評価結果を表 4.2 に示す．CUSBagging, CUSBoost_TT, CUSEasyEnsemble は, 式 (4.2.1) により各クラスタから抽出する負例の数を決める CUS を利用しているモデルである．頭に「UMAP+」の付いたモデルが，クラスタリング前に UMAP によって次元圧縮したものである．この表では，ベースとなる各手法 (UnderBagging，RUSBoost，EasyEnsemble) と提案手法を比較するため，ベースとなる手法毎に色付けを行っている．つまり，UnderBagging であれば，UnderBagging，CUSBagging，UMAP+CUSBagging の中で，スコアが高いほど濃い赤色に着色している．

結果としては，ベースとなる手法とほとんど変わらないスコアとなった．その中で，僅かではあるが，CUS 又は UMAP+CUS によって，AUC-PR，再現率が向上している．ただし，全てにおいて提案手法の F1 値が悪化しているため，ベースとなる手法の閾値を変更し，提案手法と同じ再現率とした時の適合率を比較するなど，更なる検証をする必要があり，これについては今後の課題としたい．

第5章　おわりに

5.1　まとめ

本研究では，機械学習を活用した税関不正検知において，課題の一つとされている不均衡データについて，その対処法に関する先行研究を調査した．そして，先行研究において提案された手法の中から，代表的なものについて評価を行った．評価には，訓練に要する時間を含む 9 種類の性能評価指標を用いることで，各手法を多面的かつ公平に比較した．さらに，既存手法を税関不正検知へ適用し，その効果を検証した．最後に，既存手法の精度を上げるための改善策について検討を行い，その効果を検証した．

まず，代表的な既存手法の評価においては，単体の決定木より，特定のアンサンブル学習手法の精度が高くなることを確認した．そして，税関不正検知においては，アンダーサンプリングとアンサンブル学習の組み合わせが最も良い結果となることを確認した．これは，輸入申告データの特徴量の多くが，ユニークな値の数の多い質的変数で構成されていることに起因すると考えられる．

さらに，アンダーサンプリングとアンサンブル学習の組み合わせを改善するため，既存の CUS とアンサンブル学習の組み合わせについて，改善策を検討した．そして，負例の各クラスタから抽出する負例数を変更し，各弱識別器の訓練データを均衡とする方法を提案した．また，クラスタリング前に UMAP を適用することについても提案した．これらの提案手法を，既存手法である UnderBagging，RUSBoost，EasyEnsemble に適用し，人工的に生成したデータセットにおいて効果を検証した．その結果，既存手法とほぼ変わらない結果となったが，AUC-PR，再現率は僅かに向上することが確認できた．

5.2　今後の課題

Fernández らによれば，不均衡データに対するアンサンブル学習を活用した手法について，多くの研究者がその改善のために，弱識別器の多様化に取り組んでいる [15]．そして，

Pastor らは，既存の多様化手法と不均衡データに対するアンサンブル手法の組み合わせについて検証し，その有効性を報告している．本研究で提案した手法では，クラスタからランダムに抽出することのみが，弱識別器を多様化させている．したがって，弱識別器を多様化する手法を調査することが，今後の改善策の検討にあたり参考となり得る．一方，Fernández らは，例えば予測に利用する弱識別器を動的に選択したり，弱識別器の出力の集約方法を改善するといった，弱識別器のまとめ方についての研究がほとんどされていないことを指摘している [15]．本研究で提案した手法では，単に出力の平均値を取っているだけであり，これについても検討の余地がある．

提案手法におけるクラスタリングでは，k-means を利用し，クラスタ数 k はハイパーパラメータとした．しかし，UMAP による次元圧縮後，k-means ではなく HDBSCAN を適用することにより，より良いクラスタが得られる可能性がある [23]．さらに，HDBSCAN は k-means のようにクラスタ数を決める必要がない．したがって，k-means に代わって HDBSCAN を利用することも今後検討することとしたい．また，ある程度既存手法との違いが確認できれば，輸入申告データに対しての効果検証も実施したい．

本研究では，不均衡データ解析の論文で利用されている性能評価指標によって，各モデルを評価した．しかし，各評価結果に対する考察でも触れたとおり，お互いがトレードオフの関係にある指標や，実用的でないモデルに対しても高いスコアとなる指標などがあり，各モデルの性能の優劣を一概に判断することは難しい．また，実際にモデルを導入する際には，その性能を意思決定者に理解しやすいよう説明することが求められる．よって，税関不正検知において，何を重要な指標として性能改善を行うべきなのかを検討し，どのように説明すれば意思決定者が直感的に理解しやすいかを検討することが，実用に向けての非常に重要な課題であるといえる．

また，今回は検査実施済みである輸入申告のみを対象とすることにより，不正の有無が正しくラベル付けされたデータを得ることができた．一方，未活用である検査を省略した輸入申告のデータの中には，不正があった申告も多かれ少なかれ含まれる．したがって，こういったデータを何らかの形で上手く活用することにより，精度の向上が期待できる．そのための手法として，半教師あり学習や PU 学習があげられる．これらの調査も今後の課題としたい．

謝辞

本研究を進めるにあたり，多大なご尽力を頂き，御指導を賜り，幾度となく貴重な助言を頂いた滋賀大学の松井秀俊准教授に深く感謝致します．データサイエンスという未知の分野において，研究の進め方等で非常に苦慮していたところ，密に連携を取っていただいたおかげで，ここまで研究を進めることができました．また，日本初の大学院データサイエンス研究科修士課程を試行錯誤しながら作り上げてくださった，滋賀大学の竹村彰通データサイエンス研究科長をはじめとする同研究科の先生方や，職員の方々に深く感謝致します．同研究科の第 1 期修了生として，世間から認められる立派なデータサイエンティストとなるよう，今後も探究心を持って学び続けたい所存であります．そして，日々密に連携を取り合い，意見を交換し合い，切磋琢磨した同研究科の 2019 年度入学生の皆様に深く感謝致します．困難に直面した時も，皆様の助言や励ましのおかげでなんとか乗り越えて，この修士論文を書き上げることができました．最後に，この修士課程に入学し，無事に修士論文を完成させることができたのは，財務省税関・関税局の皆様のおかげでもあります．今回，私を派遣していただくにあたり，税関として前例の無いことばかりで，非常に沢山の方々に，それぞれの業務で多忙であるにも関わらず多大なご尽力を頂きました．深く感謝致します．

参考文献

[1] Sundong Kim, Yu-Che Tsai, Karandeep Singh, Yeonsoo Choi, Etim Ibok, Cheng-Te Li, and Meeyoung Cha. DATE : Dual Attentive Tree-aware Embedding for Customs Fraud Detection. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2880–2890, New York, NY, USA, aug 2020. ACM.

[2] Jellis Vanhoeyveld, David Martens, and Bruno Peeters. Customs fraud detection: Assessing the value of behavioural and high-cardinality data under the imbalanced learning issue. Pattern Analysis and Applications, No. 0123456789, oct 2019.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830, 2011.

[4] Ronaldo C Prati, Gustavo E A P A Batista, and Maria Carolina Monard. Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior. In MICAI 2004: Advances in Artificial Intelligence, Lecture Notes in Computer Science, pp. 312–321. Springer, Berlin, Heidelberg, 2004.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, Vol. 16, No. 1, pp. 321–357, jun 2002.

[6] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. IEEE, jun 2008.

[7] Dennis L Wilson. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-2, No. 3, pp. 408–421, jul 1972.

[8] J P Basu, P L Odell, and Ivan Tomek. An Experiment with the Edited Nearest-Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, pp. 448–452, jun 1976.

[9] Jianping Zhang and Inderjeet Mani. kNN approach to unbalanced data distributions: a case study involving information extraction. Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, 2003.

[10] Charles Elkan. The foundations of cost-sensitive learning. In Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2, pp. 973–978. Morgan Kaufmann Publishers Inc., 2001.

[11] Shuo Wang and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. In 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp. 324–331. IEEE, mar 2009.

[12] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In Knowledge Discovery in Databases: PKDD 2003, pp. 107–119. Springer, Berlin, Heidelberg, 2003.

[13] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol. 40, No. 1, pp. 185–197, jan 2010.

[14] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory Undersampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 39, No. 2, pp. 539–550, apr 2009.

[15] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo Prati, Bartosz Krawczyk, and Francisco Herrera. Learning from Imbalanced Data Sets. jan 2018.

[16] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res., Vol. 18, No. 1, p. 559–563, jan 2017.

[17] Zejin Ding. Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics. 2011.

[18] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE, Vol. 10, No. 3, p. e0118432, mar 2015.

[19] Zhongbin Sun, Qinbao Song, Xiaoyan Zhu, Heli Sun, Baowen Xu, and Yuming Zhou. A novel ensemble method for classifying imbalanced data. Pattern Recognition, Vol. 48, No. 5, pp. 1623–1637, may 2015.

[20] Farshid Rayhan, Sajid Ahmed, Asif Mahbub, Rafsan Jani, Swakkhar Shatabda, and Dewan Md. Farid. CUSBoost: Cluster-Based Under-Sampling with Boosting for Imbalanced Classification. In 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), pp. 1–5. IEEE, dec 2017.

[21] P S Bradley, K P Bennett, and A Demiriz. Constrained K-Means Clustering. Technical report, 2000.

[22] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. feb 2018.

[23] Umap: Uniform manifold approximation and projection for dimension reduction — umap 0.5 documentation. `https://umap-learn.readthedocs.io/en/latest/index.html`(2021-01-16).