GCI Final Report

工学系研究科航空宇宙専攻修士一年
荒居 秀尚

2018 年 7 月 16 日

概要

本稿は、現在 Kaggle において行われている Home Credit Group の Default Risk を評価するコンペティション Home-Credit-Default-Risk に絡めて Home Credit グループにとってより価値の高いクレジットスコアリングモデルを提案するものである。本稿は、背景・導入、仮説・評価、価値提案の 3 つのセクションから構成される。最初に背景・導入の章においては、クレジットスコアリングモデルの意義やその歴史的背景、そして現在における構築の手順と構築にまつわる課題を提示する。続く仮説・評価の章においては Home Credot Group におけるクレジットスコアリングモデル構築の取り組みや Home Credit Group が抱えている課題、Home Credit Group が Kaggle のコンペティションに主催者として参加するに至った経緯に関して断片的な情報から仮説構築を試み、その課題を解決するモデルの構築を行う。また、そのモデルを仮説に基づいて選択された評価手段によって評価する。最後の価値提案の章においては提案するモデルが持つ価値を Home Credit Group にもたらす影響から概算し、既存の信用機関におけるクレジットスコアリングモデルの構築の費用などを参照したうえで価格提示を行う。

1 背景・導入

1.1 クレジットスコアリングの意義

銀行や信用機関における信用リスク管理業務は単なる貸出審査の効率化以上の意義があると考えられる。貸出業務は金融機関の業務の中でも大きな割合を占めるが、そこで重視されるのはリスクとリターンのバランスである。貸出にあたっての不確定性が大きく完全な資金回収を行うことができないことは大きな痛手となる一方で、確実性を求めて一部の優良な取引相手や確実な資金回収が見込まれる相手だけと取引をすることは、銀行・金融機関の利益にあまりつながらない。したがってその中間をとり、リスクとリターンがバランスするような点において取引をするか/しないかを定めるのは非常に重要と言える。この決定は一般には銀行・金融機関のリスク選好性によって変わるものであるが、その決定に大きな影響を及ぼすのが信用リスク管理である。

信用リスクの管理は「貸出資産の損失可能性を事前に推算すること」と言い換えることもできるが [1]、これにより貸出資産の価値を評価することができ、金融機関それ自体の健康状態を管理することができるともいえる。金融機関の健康状態の悪化はしばしば経済的な不安要因となり社会を大きく揺るがすことにもつながりかねない。近年では、米国に端を発した信用危機が世界的

な経済不安を引き起こすなどの例もあったが、これも信用リスク管理の失敗によるものであり、この業務の重要性を端的に表した一例と言える。クレジットスコアリングは上記のような信用リスク管理業務のなかでも大きな重要度を持った業務である。信用リスクの推算は貸出を行った場合の予想損失 (EL) の推算とも言い換えられるが、予想損失の構成パラメータは、デフォルト時貸出残高 (EAD)、デフォルト率 (PD)、デフォルト時損失率 (LGD) の積として表される。式として表すと

$$EL = EAD \times PD \times LGD$$

となる。このうちクレジットスコアリングが行うのはデフォルト率 (PD) の推算である。この推算を誤ると、実際には大きなリスクを抱えている案件のリスクを過小評価してしまったり実際には優良な債務者に対して貸出を行わない、といった意思決定に繋がる恐れがあり非常に重要である。

1.2 クレジットスコアリングの歴史的背景

クレジットスコアリング自体の登場は非常に古く、1950 年代に開発され始め 1960 年代に実用化されたと言われている [2]。これらは、登場当初は上述のような理由は主ではなくどちらかといえば現場の負担を軽減する目的で導入されたと言われているがその後急速に普及した。

米国では過去数十年にわたってクレジットスコアリングが消費者の生活の非常に大きな部分を占めてきた経緯がある。クレジットスコアが生活に影響を与える範囲は広く、元々はローンの貸出などの指標として用いられていたものの、住宅の入居判断や就職などにおいてもクレジットスコアの良し悪しが関わってくるようになり現在では生活の多くの部分を支配し格差を固定化する要因の一つとなっているとも言われる。米国で普及するクレジットスコアリングは大きく 3 つの企業による指標が用いられることが多い。その 3 企業は TransUnion、Equifax、Experian であるがこれらは FICO スコアという指標の計算方法をベースに独自のクレジットスコアの算出をしている [3] が大まかには以下のような内訳であると言われている。

- 35% が過去の支払い履歴
- 30% が現在の負債
- 15% が信用履歴の長さ
- 10% が最近の融資問い合わせ額
- 10% がアカウントの種類/個数

また、クレジットスコアを改善させるためのベストプラクティスといったものもよく伝え聞かれており以下のようなものが挙げられている。

- クレジット限度額の 65%-75% には手をつけないようにすること。多すぎるとリスクになり、少なすぎると信用機関にとって良くわからない人になる。
- 昔のアカウントを close しないで open なままにしておくこと、誘惑に負けてしまいそうならカードを捨てなさい。
- 新規アカウント開設は控えめに。企業や貸し手が信用情報を参照するたびにクレジットス

コアは下がります。

- 延滞や延滞期限をできるだけ残さないようにすること。
- 安易な解決策に飛びつかないように。詐欺の場合もあります。

1.3 現在における構築の手順・課題

クレジットスコアリングモデルは各金融機関の生命線であるため具体的なアルゴリズムや説明変数が公開されている例はあまり存在せず、断片的な情報を組み合わせて推測する必要がある。また、信用機関は目的に応じて複数のクレジットスコアリングモデルを使い分けていると考えられるため [4]、よく用いられていると考えられる手法をその背景などに着目しながら紹介する。また、クレジットスコアリングに用いられる説明変数はその国や地域における法規や産業構造の違いなどから異なる可能性が示唆される [5]。

1.3.1 クレジットスコアリングの手法

現在のクレジットスコアリングはその用例に応じて複数の手法を使い分ける試みがなされていると考えられる [4]。以下はそれらについてと、その用法に関しての説明である。

- スコアカード

経験的な指標に基づきある条件を満たしているときは点数を加算することで、合計点数をクレジットスコアとするモデルである。例としては、勤続年数 1 年未満は 10 点、1-3 年は 30 点、3-10 年は 50 点、10 年以上は 70 点と言った具合である [6]。非常に単純な指標であるため理解は容易であるが、経験的な指標であるため意思決定の際の強い根拠とするには弱く、また現実の複雑な状況に即した判断にも使いづらいという難点が存在する。

- 判別分析

線形分離モデルを用いてデフォルト先と非デフォルト先をより良く区分する線を決定する手法である。線形手法であるため、複雑な境界を表現できないという点で現実の複雑な状況に即した判断には使いづらく、直感的な理解もスコアカードに比べると容易ではない。そのため、意思決定の際にも強い根拠として用いることが難しい。

- ロジスティック回帰

一般化線形モデルを用いてデフォルト確率を推算することができるため、意思決定においては使いやすい。一方で、表現力は線形判別分析とおよそ同程度である。直感的な理解のしやすさも判別分析と同程度と言える。

- 決定木

確率の推算にも用いることができるため意思決定においては使いやすい。非線形の決定境界を表現できる一方、過学習しやすいという欠点もあり複雑な状況に即した判断にはやや使いづらい。一方、判断の根拠を階層的な分岐で表現でき理解のしやすさは判別分析やロジスティック回帰よ

りよいといえる。

- ニューラルネット

中間層の複雑さを増すことでより複雑な境界を表現できる一方で、説明性に劣る面があるため、意思決定においても使いづらいという側面が存在する。

- カーネル SVM

カーネル関数を用いて高次元空間に特徴を写したうえで線形分離をする手法で、数年前までニューラルネット以上の成績を出すと言われていた一方、説明性に劣るため、意思決定の場では使いづらいという側面がある。

- アンサンブル手法

勾配ブースティングやニューラルネットと決定木のブレンディング・スタッキングなど、近年 Kaggle などのコンペティションにおいても隆盛を誇っている手法である。シングルモデル以上の成績を示すこともある一方で説明性は低く意思決定の場で決定打とはならないという現状があると考えられる。

1.3.2 説明変数について

説明変数の選択に関しては、スコアリングの具体的なアルゴリズム以上に企業の秘密に関わるため、公開されている例はないが断片的な情報からおおよそどの企業も用いていそうな説明変数の推測は可能である。一方、その他の部分は企業ごとのスコアリングにおいて大きな特徴となりうる一方で推測は難しくその個数もスコアリングを行う企業によって異なることが予想される。説明変数の選択についてもアルゴリズムの選択同様、用途によって分けていると考えられるため、参照できる資料をもとに考察を行う。

まず、説明変数の個数についてであるが、一般的に 50 以上になることはないようである [5][2]。これは、説明変数の増大に伴いモデルの複雑さが増し、説明性の低下を招いたり保守管理のコストが増すといった側面があり、説明変数を増やすことによる表現力の向上との兼ね合いから考えてもこれ以上の数にはしづらいという背景が存在する。また、近年はビッグデータの活用などにより複雑なモデルに大量のデータを投入するような PD 推定の取り組みもあるものの、現状ではいくつかの"スマートに"加工された説明変数を用いることで PD 推定を行う手法が主流であり、説明変数の個数のおおよその値としては 9 50 程度と見積もることができる [5][7]。説明変数の種類に関しては、各種金融機関に対する国土交通省住宅局の 2011 年の調査に挙げられたものを参考とした [7]。この調査では住宅ローン審査において金融機関が審査項目に入れることが多かった項目を項目別割合として結果にあげている。

また、[8] などでは、変数選択により (1)「勤続年数」(2)「自己資本比率」(3)「合算年収倍率」(4)「借入期間」(5)「事業主フラグ」(6)「小企業フラグ」(7)「中企業フラグ」(8)「地域別 CI 成長率」(9)「地域別失業率」の 9 変数に絞って PD 推定を行っている。いずれにしても、生活基盤の安定性などを評価する指標 (勤続年数・年収・雇用先規模・業種・健康状態など)、過去の信用情報

審査項目	回答数の割合
完済時年齢	91.90%
返済負担率	90.10%
借入時年齢	89.90%
勤続年数	88.50%
年収	88.10%
健康状態	88.10%
担保評価	85.90%
連帯保証	85.10%
債務状況・返済状況	84.90%
融資可能額 (購入時)	84.70%
融資可能額 (借り入れ時)	83.30%
金融機関の営業エリア	80.20%
雇用形態	63.10%
申込人との取引状況	59.70%
国籍	47.60%
業種	40.20%
雇用先の規模	26.60%
家族構成	26.00%
所有資産	22.70%
性別	16.50%
その他	9.70%

住宅ローン審査において民間金融機関が重視する項目

履歴 (債務状況・返済状況、自己資本比率)、将来性 (完済時年齢、借入時年齢、借入期間) などを説明変数に取り入れている一方で、コンプライアンス上問題になりかねない説明変数 (家族構成・性別・国籍など) は審査項目として使われづらい傾向があると考えられる。

1.3.3 金融機関の抱える課題

クレジットスコアリングに関連して金融機関が抱える課題はいくつかに分けられる [4]。

- モデルが複雑なため、作成過程や計算過程がブラックボックス化してしまう
- モデルの精度に不安があり、モデルの見直しの必要がある
- モデルの見直しのたびに、コンサルティングやシステム改修の費用がかかる

といったことが挙げられている。これらに関して金融機関により挙げられている対策は以下のようになる。

- 研修の実施

- 運用目的にあわせて複雑さと説明性のトレードオフを考慮したモデルの再構築
- 最新の機械学習手法の利用
- テンプレートの利用

また、この他に [5] などに挙げられている、詐欺のリスクも大きな課題となっているようである。詐欺を試みる者は虚偽の申告をする場合も多く、データには現れづらいと考えられる。

2 仮説・評価

2.1 Home Credit Group におけるクレジットスコアリングモデル構築の取り組み

Home Credit Group におけるクレジットスコアリングは Kaggle のコンペティションのディスカッションにおいて主催者である Kirill Odintsov 氏が語っている内容から部分的に推察できる [5]。以下はそれらのディスカッションの内容を要約したものである。

- AUC 0.77 は現行のモデルよりはやや低いが抜かれるのは時間の問題である。それは、ビジネスにおける制約上用いることが出来ないデータを今回のコンペティションでは数多く用意しているからである。
- Home Credit Group において用いられているモデルはロジスティック回帰や決定木などの単純なものもあるし、より高度な XGB などの勾配ブースティング法やニューラルネットを使ったものも存在する。
- 既に多くの融資を受けている人にお金を貸すことは企業としては利益になると考えられるので積極的に行うべきことであるが、顧客を過剰に借金を抱えた状態にしてしまう可能性があるので PD が小さいと予測されていたとしても避けたいことである。
- 100 以上の特徴を加えてモニタリングするよりも、50 くらいの特徴でやや低いくらいの精度を出せるほうが良い。
- 最終的にはより簡単なモデルを使うにしても一度は複雑なモデルを試すことを行っている。
- モデルの再構築時には古いモデルを通ってきた母集団で学習を行うため、そこには含まれていなかったような人が新規申し込みをしたとして本来は落とされるべきであるにもかかわらず審査を通過してしまう可能性があり大きな問題となっている。
- 実際のモデルでは、性別・結婚状態・子供の数など、コンプライアンス的に使えない変数がある。
- その時点での景気などの情報を使えば推論の精度は良くなるがそれは入れられない。
- Home Credit Group のデータサイエンスチームが用意しているモデルは多数ある。

1. 顧客が最適なローン額を選ぶのを助けるモデル
2. 抱えているリスクから利子率を算出するモデル
3. データの収集戦略を最適化するモデル
4. ローンの提案をするモデル
5. 大規模な詐欺行為を調査するモデル
6. 顧客のクレジットカード制限の増減を決めるモデル

- ローンの契約は国や地域によって異なると考えられる (法的要請や産業構造の違いなどから)。応募用フォームは確かに異なるが現在データサイエンスチームが国ごとに大きく異ならないようにしているところである。
- 今回、Kaggle でコンペティションを開催したのは実際の業務において用いるためではなく、Kaggler のデータ分析のやり方などを見るためである。

2.2 Home Credit Group がコンペティション主催に至った理由

Home Credit Group がコンペティションに求めているものは上述の通り、Kaggler のデータ分析のやり方そのものであり、実際の業務を改善させるようなモデルではないと考えられる。しかし、データ分析のやり方が求めるものだったとして、そのやり方を学ぶことで業務を改善させるようなことができると考えているからこそ総額$ 70,000 の賞金を設定するほど大きなコンペティションを開催する事になったと考えられる。すなわち、Home Credit Group ではデータ分析業務やリスク管理業務そのものが問題となっているという仮説が建てられる。言い換えれば、スコアリングのモデルの精度が低いといった問題というよりも、スコアリングのモデルの作成の仕方などが問題となっていると考えることができる。

この観点で Kirill 氏のディスカッションにおける発言を見直すと、モデルの作成に絡めて大きな課題意識を持っていることが伺える発言をしていることがわかる。

> We rebuild our models quite often. Every time we rebuild a model we can do it only on approved clients by the old model (we have no target for rejected clients). When we implement new model we stop using the old one, but the new model can start approving people who previously would be rejected by the old model not because they are good but simply because the new model did not learn that they are bad because they were not in the dataset for the new model. So for example if your old model would see that the people with "characteristic A" are really bad it would reject almost all of them - only those with best other characteristic would have a chance to be approved. The next model you will build on clients approved by the old model. Thus you would have small amount of people with "characteristic A" and they would not seem so bad, but in reality they are you just can't observe it in the data because the old model took only the best ones with "characteristic A".

これに絡めた発言を同氏は複数回しており、この問題を解決するのは非常に難しいということを認めている。これらの発言をまとめると、Home Credit Group が抱えている課題意識は以下のようになる。

- モデルを再構築する際に、モデルの学習に使うデータは古いモデルによる予測を元に与信をすると意思決定した人たちのものである。
- 古いモデルが与信をしないと選択した対象は何らかの特徴が悪かったために落とされたと考えられるが、新しいモデルを学習させるのに使うデータにはそれらの特徴が"悪い"とされるようなデータは含まれていないか、含まれていたとしてもその他の特徴が"良い"とさ

れたために審査を通過した人たちである。
- このようなデータで学習を行った場合、前回の審査では落とされてしまったような人も新しいモデルによる審査は通過する可能性がある。
- なぜならば、前回の審査で申込者を落とすのに主要な役割を果たした特徴に関して、新しいモデルの学習用データには圧倒的に偏りが生じているためである。

この他の点に関しても、Home Credit Group はいくつかの課題を抱えていると考えられるが、大きなものとしてはこの点に尽きると考えられる。

2.3 課題を解決するアイデア

Home Credit Group の抱える課題は上述の通りであるが、これを解決するのは Kirill 氏のいうように簡単な話ではない。しかし、いくつか解決に繋がりそうなアイデアは挙げられる。

1. 新しいモデルの学習において、古いモデルによって審査落ちしてしまった対象群に関しても学習データとして用いる。その際、そのようなデータに関してはターゲット変数がデフォルト (今回であれば 1) であるとして扱う。
2. モデルの説明性を良くし、審査に落ちた人に関してその審査に関する支配的な特徴を特定できるようにする。審査を段階的なステップを踏むことにより、1 段回目の審査で古いモデルで落ちてしまうような特徴を持つ人をふるい落とし、2 段階目で新しいモデルを用いた審査を行う。モデルが更新されるごとにステップを増やす。
3. スコアリングモデルへの依存度を下げ、あくまでシグナルを出す程度のものに留める。すなわち最終的な審査は人手で行う。
4. スコアリングモデル自体は人間が確認できる程度の複雑さにし、機械による選別と人間による選別の 2 段階審査を行う。3 のアイデアに近いが、機械的な作業の割合の問題である。
5. 1 と 2 の折衷案として、古いモデルにおいて審査において落とすと決定したときの支配的要因を特定し、学習用データにおいてそのような特徴が"悪い"とされるデータを人工的に創りだす。この際、その人工データに関してはターゲット変数をデフォルトであるとして扱う。

上記のアイデアには当然のことながら長所もあれば短所も存在する。今回は特に短所を挙げていく。

- 新しいモデルの学習において、古いモデルによって審査落ちしてしまった対象群に関しても学習データとして用いる。その際、そのようなデータに関してはターゲット変数がデフォルト (今回であれば 1) であるとして扱う。

古いモデルによって審査落ちしてしまった対象群は実のところ 2 通り存在すると考えられる。すなわち、PD が高いという予測が立った人たちである可能性と、リスクとリターンのバランスを考慮した時に PD はそれほど高くないものの、その他の要素を考慮した際に魅力的な申込者と判断されなかった可能性が考えられる。これらを一緒くたに扱ってしまうと、PD を予測するモデルに PD はそれほど悪くなかったデータまで混入することになり精度の悪化を招いてしまう。

- モデルの説明性を良くし、審査に落ちた人に関してその審査に関する支配的な特徴を特定できるようにする。審査を段階的なステップを踏むことにより、1 段回目の審査で古いモデルで落ちてしまうような特徴を持つ人をふるい落とし、2 段階目で新しいモデルを用いた審査を行う。モデルが更新されるごとにステップを増やす。

多段階モデルの欠点は二点ある。1 つはそもそも古いモデルを使っている点にある。モデルを更新する必要があったということは、古いモデルに何らかの欠陥が存在した可能性もあるため、そのようなモデルを一段階目で使用するのはデータの分布をゆがめかねず危険である。また、モデルの更新のたびに新しいステップが増えるためモデルの複雑性が増大していってしまう点にある。結果としてモデルの管理コストやヒューマンエラーのリスクが増大することにつながりかねない。

- スコアリングモデルへの依存度を下げ、あくまでシグナルを出す程度のものに留める。すなわち最終的な審査は人手で行う。

人間による審査は負担を増大させヒューマンエラーの危険性を飛躍的に増大させるほか、人件費の増加も招くことになりスコアリングの取り組みに関しては大きく逆行することになる。

- スコアリングモデル自体は人間が確認できる程度の複雑さにし、機械による選別と人間による選別の 2 段階審査を行う。3 のアイデアに近いが、機械的な作業の割合の問題である。

機械に扱えるデータの複雑さに制限をかけすぎてしまい、人間への負担が大きくなることが考えられる。最終的な判断はしばらくは人間に委ねられ続けるにしろ、やはりスコアリングの取り組みに逆行するアイデアである。

- 1 と 2 の折衷案として、古いモデルにおいて審査において落とすと決定したときの支配的要因を特定し、学習用データにおいてそのような特徴が"悪い"とされるデータを人工的に創りだす。この際、その人工データに関してはターゲット変数をデフォルトであるとして扱う。

1 つの特徴だけではなく、複数の特徴が複合的に影響して"悪い"という評価を作っていた場合に人工データの生成時に漏れが生じる可能性がある。

2.4 提案する解決策

以上の考察を踏まえたときに案 1 は改善の余地があるように思われる。つまり古いモデルにおいて PD が高いとされて審査落ちしたグループについてそのデータを学習に用いることが挙げられる。ただしこの群に関して一律に目的変数がデフォルト (1) であるとして学習を行うとデータの分布を損なってしまうため、サンプリングなどを用いて適切に評価する必要がある。したがって今回提案する解決策は以下のようなものである。

- 古いモデルにおいて PD が高いとされたレコードについては、新しいモデルの学習時にも用いる。
- 古いモデルにおいて PD が高いとされたレコードを新しいモデルの学習データとして用い

るときは、古いモデルの予測 PD に従いサンプリングを行いサンプリングされたデータを古いモデルにおいても PD は低いとされていたデータと結合して用いる

この方法でも、古いモデルに依存した部分が出てきてしまうため、> モデルを更新する必要があったということは、古いモデルに何らかの欠陥が存在した可能性もあるため、そのようなモデルを一段階目で使用するのはデータの分布をゆがめかねず危険である。

この指摘に関しては対応できていない。

2.5 実験

以上の議論を踏まえ、提案手法の性能を評価する実験を行った。データセットに関しては、今回 Home Credit Group から、Kaggle のコンペティション用に提供されているデータのうち、ターゲット変数が明らかになっている application_train を用いた。また、その他の外部リソースや過去の申し込み情報などは今回の「提案手法の評価」という点では必要ないため、使用しなかった。また、説明変数も今回は提案手法とそれを用いなかった場合の差分を見るためだけのため、9 個程度に絞って行った。変数の選択に関しては、LightGBM を用いたモデル [9] による変数重要度と [8] に挙げられた変数などを参考に、

1. DAYS_EMPLOYED(勤続年数)
2. AMT_CREDIT/AMT_INCOME_TOTAL(自己資本比率に類似)
3. AMT_CREDIT/AMT_GOODS_PRICE
4. AMT_CREDIT/AMT_ANNUITY(借入期間に類似)
5. REGION_POPULATION_RELATIVE(Target Encoding をして地域別のデフォルト率指標とする)
6. DAYS_BIRTH
7. DAYS_EMPLOYED/DAYS_BIRTH
8. DAYS_ID_PUBLISH
9. ORGANIZATION_TYPE(Target Encoding により職種別デフォルト率指標とする)

今回はあえて EXT_SOURCE_とついた指標については用いなかった。

データセットについては、古いモデルの学習用データ、古いモデルにより分類されるデータ (新しいモデルの学習データにもなる)、提案手法の性能確認用データの 3 つに分けた。全体の流れとしては以下の通りである。

1. application_train のデータから 9 変数を抽出し、3 つのグループに分類する。それぞれ fold1, fold2, fold3 とする。
2. Old Model を fold1 データで学習させる。今回はモデルとして RandomForestRegressor を用いる。
3. Old Model で fold2 データに関して PD の予測を行う。予測された PD に関して閾値以上であったデータは審査落ちデータとして扱う。
4. New Model1 に関して、審査落ちデータは用いずに学習を行う。これは現状 Home Credit Group で行われている取り組みに対応する。New Model1 も RandomForestRegressor

とする。

5. 審査落ちデータに関してはステップ 3 で予測された PD をもとにサンプリングを行い審査落ちデータが審査を通っていたと仮定した場合のデータセットを生成する。なお、サンプリングの個数はハイパーパラメータとなる。
6. ステップ 3 で審査を通過したデータについては fold2 の target をそのまま、審査落ちとされたデータについては 5 でサンプリングをした結果を target として用いる。これによりサンプリングの回数分のデータセットが作成される。
7. New Model2 に関して、6 で作成したデータセットで学習を行う。この際にサンプリングの回数分の回帰木を作成する。モデルは 4 で用いたものと同じとする。
8. New Model1、New Model2 に関して fold3 のデータで検証を行う。

2.6 評価に関して

クレジットスコアリングのモデルの評価においてよく用いられるのは、AR(Accuracy Ratio) と呼ばれる指標の様である [1]。これは、序列性能を図る指標でありほとんど、ROC-AUC と等価である。しかし、今回は序列性能が目的というよりは、New Model の学習時に学習用データセットに、前のモデルで PD が高いとされた群を含むことで、新規加入者のうち前のモデルでは落とされてしまったような人が入り込むことを防ぐのが目的である。

したがって今回の実装においては評価は、以下のような方法で行う。

1. fold2 データ全体を用いて学習したモデルで fold3 データを予測する。pred3 と呼ぶことにする。
2. New Model1(審査落ちデータを含まない) の fold3 データに関する予測を pred1 とする。
3. New Model2(提案手法) の fold3 データに関する予測を pred2 とする。
4. pred3 と pred2 の差分が大きいデータ (すなわち審査落ちデータを用いなかったことによって PD の予測を誤ったことに相当する) に関して pred1 が一定の割合以上の PD を予測できていた割合を計算する。

2.7 実装

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sys
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import RandomForestRegressor
import numpy.random as random
```

2.7.1 Step 1 データの整形

```
class TargetEncoder:
    def __init__(self):
        self.encoder = None

    def fit(self, cat, target):
        colname_cat = cat.name
        colname_target = target.name

        concat = pd.concat([cat, target], axis=1)
        self.encoder = concat.groupby(colname_cat)[colname_target].mean()

    def transform(self, cat):
        target = cat.map(self.encoder)
        return target

    def fit_transform(self, cat, target):
        self.fit(cat, target)
        encoded = self.transform(cat)
        return encoded

df_path = "../data/application_train.csv"
df = pd.read_csv(df_path)
df = df.dropna(subset=["AMT_GOODS_PRICE", "AMT_ANNUITY"])

df["DAYS_EMPLOYED"] = df["DAYS_EMPLOYED"].map(lambda x:x if x != 365243 else 0)
df["CREDIT_INCOME_RATIO"] = df["AMT_CREDIT"] / df["AMT_INCOME_TOTAL"]
df["CREDIT_GOODS_RATIO"] = df["AMT_CREDIT"] / df["AMT_GOODS_PRICE"]
df["CREDIT_ANNUITY_RATIO"] = df["AMT_CREDIT"] / df["AMT_ANNUITY"]
df["EMPLOYED_BIRTH_RATIO"] = df["DAYS_EMPLOYED"] / df["DAYS_BIRTH"]

te = TargetEncoder()
df["REGION_TARGET_ENCODED"] = te.fit_transform(
    df["REGION_POPULATION_RELATIVE"], df["TARGET"])
df["ORGANIZATION_TARGET_ENCODED"] = te.fit_transform(
    df["ORGANIZATION_TYPE"], df["TARGET"])

columns_to_use = ["DAYS_EMPLOYED", "CREDIT_INCOME_RATIO", "CREDIT_GOODS_RATIO",
```

```
                    "CREDIT_ANNUITY_RATIO", "REGION_TARGET_ENCODED",
                    "DAYS_BIRTH", "EMPLOYED_BIRTH_RATIO",
                    "DAYS_ID_PUBLISH", "ORGANIZATION_TARGET_ENCODED"]

X = df[columns_to_use].values
y = df["TARGET"].values

skf = StratifiedKFold(n_splits=3)
for train_idx, test_idx in skf.split(X, y):
    train_tmp = X[train_idx]
    y_train_tmp = y[train_idx]
    Xfold3 = X[test_idx]
    yfold3 = y[test_idx]

skf2 = StratifiedKFold(n_splits=2)
for train_idx, test_idx in skf2.split(train_tmp, y_train_tmp):
    Xfold1 = train_tmp[train_idx]
    yfold1 = y_train_tmp[train_idx]
    Xfold2 = train_tmp[test_idx]
    yfold2 = y_train_tmp[test_idx]
```

Xfold1, Xfold2, Xfold3 がそれぞれ fold1, fold2, fold3 データに対応し yfold1, yfold2, fold3 はそれらのターゲット変数である。

2.7.2 Step 2 Old Model の学習

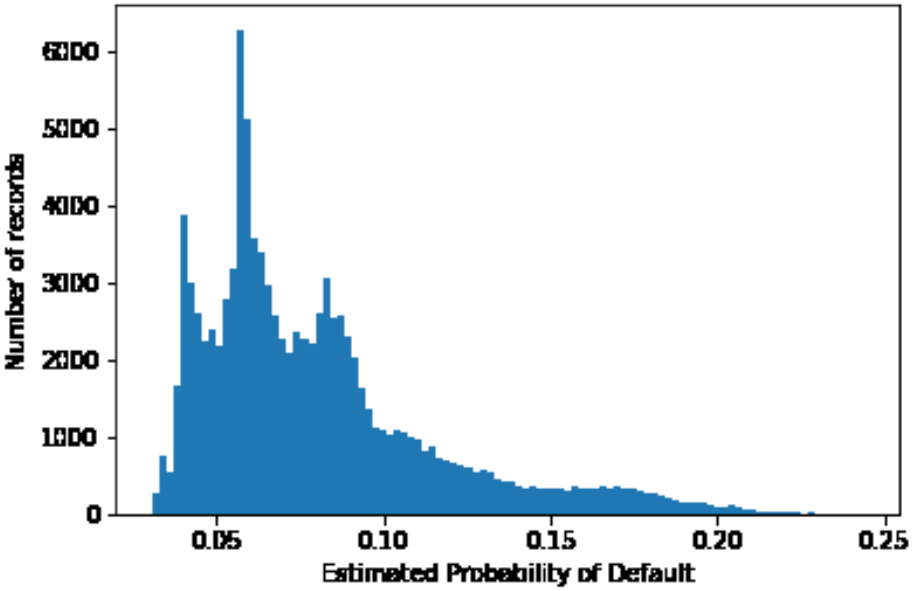
```
old_model = RandomForestRegressor(max_depth=5, n_estimators=100, n_jobs=-1)
old_model.fit(Xfold1, yfold1)
```

2.7.3 Step 3 Old Model による PD 予測

```
ypred_old = old_model.predict(Xfold2)
plt.hist(ypred_old, bins=100)
plt.xlabel("Estimated Probability of Default")
plt.ylabel("Number of records")
plt.show()

use_idx = np.where(ypred_old < 0.15)[0]
discard_idx = np.where(ypred_old >= 0.15)[0]

Xnew = Xfold2[use_idx, :]
```




```
ynew = yfold2[use_idx]

Xdiscard = Xfold2[discard_idx, :]
ydiscard = yfold2[discard_idx]
ypred_discard = ypred_old[discard_idx]
```

今回は閾値を 0.15 として計算した。

2.7.4 Step 4 New Model1 の学習

```
new_model1 = RandomForestRegressor(max_depth=5, n_estimators=100, n_jobs=-1)
new_model1.fit(Xnew, ynew)
```

2.7.5 Step 5-7 データセット作成~New Model2 学習

```
class SamplingRegressor:
    def __init__(self, data, ypred, base_estimator, nsample=200):
        self.nsample = nsample
        self.clfs = [base_estimator for i in range(nsample)]
        self.discarded = data
        self.samples = [random.binomial(1, ypred) for i in range(nsample)]
        self.X = None
        self.y = []

    def fit(self, X, y):
        self.X = np.vstack([X, self.discarded])
```

```
        self.y = [np.hstack([y, s]) for s in self.samples]
        for i, clf in enumerate(self.clfs):
            clf.fit(self.X, self.y[i])
            percentage = (i+1) / self.nsample * 100
            sys.stdout.write(f"\r{percentage:.2f} percent finished")
        print()

    def predict(self, X):
        preds = np.zeros(X.shape[0])
        for clf in self.clfs:
            preds += clf.predict(X)
        return preds / self.nsample

rf = RandomForestRegressor(max_depth=5, n_estimators=100, n_jobs=-1)
new_model2 = SamplingRegressor(Xdiscard, ypred_discard, rf, 1000)
new_model2.fit(Xnew, ynew)
```

2.7.6 Step 8 **検証**

```
validation_model = RandomForestRegressor(max_depth=5, n_estimators=100, n_jobs=-1)
validation_model.fit(Xfold2, yfold2)

pred1 = new_model1.predict(Xfold3)
pred2 = new_model2.predict(Xfold3)
pred3 = validation_model.predict(Xfold3)

idx = np.where(pred3 - pred1 > 0.1)[0]
print((pred2[idx] > 0.15).mean())
```

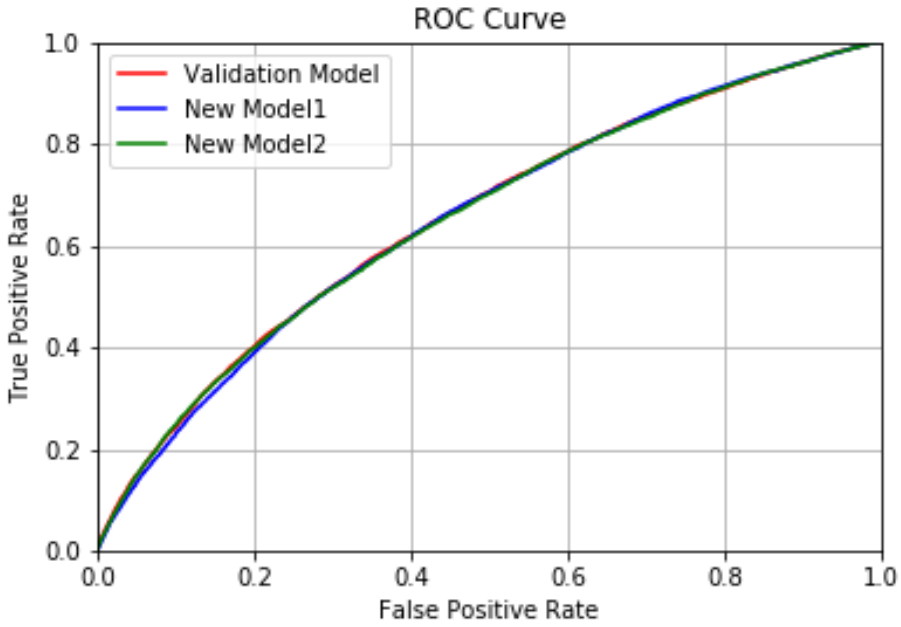
これにより Old Model では審査落ちしていて New Model1 では PD が低く出ていたデータのうち **98.8%** が提案手法では PD が高い (閾値以上) と予測されることがわかった。

なお、確認のため ROC 曲線を描いてみたのが以下の図である。AUC では、Validation Model が 0.653, New Model1 が 0.649, New Model2 が 0.651 となった。

3 価値提案

最後に提案したモデルについて、価格設定を行う。本モデルには **$ 40,000** の価格を設定する。この価格の根拠は以下のとおりである。

まず、Home Credit Group が Home-Credit-Default-Risk のコンペティションに設定した賞金は総額$ 70,000 であり、一位の賞金は$ 35,000 である。この価格設定は、コンペティションに



対して Home Credit Group がかけている期待を反映したものであると言える。先に述べたとおり、Home Credit Group がこのコンペティションに求めているものは、Home Credit Group が抱えている課題を解決するデータ分析のやり方のヒントであって提出されるモデルそのものではない。すなわち、実務に直接使えるものではなく、その業務を改善するヒントに$ 70,000 という額を設定していると考えられる。これは将来にわたってのクレジットスコアリングモデル構築費用の一部の額を積分した値と捉えることができる。

また、同業である富士通総研ではクレジットスコアリングモデル構築に$ 50,000 以上という価格設定をしている [4]。これを考えて、クレジットスコアリングモデルの相場はおよそ$ 30,000 - $ 100,000 程度であると予測した。この上で今回のスコアリングモデルは Home Credit Group が抱える課題にクリティカルに作用する解決策を含むもののその他の部分での手法的目新しさはなく、運用中に改善を多く施すことになると考えられることから、スコアリングモデルの価格の下限に近い$ 40,000 を設定した。

参考文献

[1] 尾藤剛. ゼロからはじめる信用リスク管理銀行融資のリスク評価と内部格付制度の基礎知識. 一般社団法人 金融財政事情研究会, 2011.

[2] 日向野幹也. perspectives.pdf. https://www.yu-cho-f.jp/research/old/pri/reserch/monthly/2000/147-h12.12/perspectives.PDF.

[3] Catherine J. Byerly. The truth about your annuity and your credit. https://www.annuity.org/2015/10/15/truth-on-annuity-and-credit/.

[4] 株式会社富士通総研第二コンサルティング本部ビジネスアナリティクス事業部. クレジットスコアリングモデル構築/運用支援.

http://www.fujitsu.com/downloads/JP/archive/imgjp/group/fri/service/credit_scoring.pdf.
[5] KirillOdintsov. Welcome note for home credit. https://www.kaggle.com/c/home-credit-default-risk/discussion/57054.
[6] クレジットカード審査のスコアリング. https://www.woshiru.com/creditcard/kisochishiki/shinsa/shiny
[7] 国土交通省. 平成 23 年度民間住宅ローンの実態に関する調査の結果について. http://www.mlit.go.jp/report/press/house01_hh_000047.html.
[8] 各務 和彦 奥村 拓史. 階層ベイズ・モデルによるクレジット・スコアリング・モデル: 住宅ローンコンソーシアム・データへの応用. https://www.terrapub.co.jp/journals/jjssj/pdf/4201/42010025.pdf.
[9] Pooh. Fork_of_fork_lightgbm_with_simple_features. https://www.kaggle.com/poohtls/fork-of-fork-lightgbm-with-simple-features.