Vol.2021-SE-207 No.13
2021/3/1

日次更新される機械学習モデルの品質確認フレームワーク

内藤裕暉[1]　指田晋吾[2]　今村光良[2]　岸知二[1]

概要：機械学習モデルの品質確認は重要だが困難な課題である．我々は日次更新される機械学習モデルを運用する状況を想定し，モデルそのものの品質を捉えるモデル作成者の内部視点からの確認を，モデル利用に関わる品質を捉える利用者の外部視点の確認によって補完する品質確認フレームワークについて検討している．金融時系列予測モデルを例に品質確認フレームワークの有効性や課題について検討する．

A Quality Evaluation Framework for Machine Learning Model Updated on a Daily Basis

HIROKI NAITO[†1]　SHINGO SASHIDA[†2]
MITSUYOSHI IMAMURA[†2]　TOMOJI KISHI[†1]

1. はじめに

機械学習に代表される AI システムの活用が広まっているがそのモデル（ML モデル）や，それを用いたシステム（ML システム）の品質確認には様々な課題がある．例えば ML システムの技術的負債についての議論では，その特徴や従来のソフトウェアシステムとの違いについて指摘されている[4]．そうした中，ML システム・ML モデルに対するテスト方法やテスト設計の提案などが多数なされている[8]．

また ML システムはその適用分野や運用形態によって，求められる品質やその確認方法が異なることも報告されている[2][11][15]．従ってモデルの特性，分野，運用形態，また組織の成熟度などに応じた品質の議論や品質確認の手法の検討が重要となる．

我々は日々更新される ML モデルを運用するという状況を想定し，その品質を確認するフレームワークについて検討している．ここでは ML モデルそのものの品質を確認する ML モデル開発チームと，ML モデルの利用時の品質を確認する運用チームが独立した視点で確認を行う．本稿では特に，運用チームによる確認手法に焦点をあて，差分テストに基づく品質確認の基本的な考え方と，金融分野での事例に対して試験適用してみた結果を踏まえ，品質確認フレームワークの妥当性や課題について議論する．

2 章では本研究の問題設定について述べ，3 章では提案する品質確認フレームワークの基本的な考え方について述べる．4 章では本品質確認フレームワークの事例に対する試験適用について説明する．5 章では関連する議論を行い，6 章で本稿を結ぶ．

2. 問題設定

本研究では，以下のような ML モデルの運用を想定する．

- ML モデルの開発チームと，その ML モデルを利用する運用チームから構成される．
- 開発チームは日々変更するデータに基づいて ML モデルを生成するとともに，ML モデルの実現技術に依存した品質確認を行う．
- 運用チームは，ML モデルを業務に活用するため，そのモデルを運用に使って問題ないかについて短時間で確認を行った上でその ML モデルを利用する．
- ML モデルは将来の予測に使われるもので，その出力に対する明確な正誤判断はできない．

図 1 は，想定する ML モデルの生成と運用を模式的に示したものである．ある日（d とする）において，開発チームは一昨日（d-2）までのデータを学習データとして ML モデルを生成する．運用チームはその ML モデルに昨日（d-1）のデータを入力して出力を得て，それを業務に活用する．

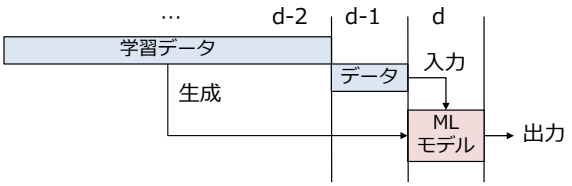

図 1　日次の ML モデル生成・運用

上記の設定の下，ML モデルの品質の効果的な確認方法を検討する．今回は運用チームによる確認に焦点をあて，開発チームから渡される ML モデルをその日の運用に利用

1 早稲田大学
Waseda University
2 野村アセットマネジメント株式会社
Nomura Asset Management Co. Ltd.

するに際し，その品質について短時間で評価を行うための方法を検討する．

3. 品質確認フレームワーク

本章では提案する品質確認フレームワークについて説明する．

3.1 基本的考え方

提案する品質確認フレームワークは以下の考え方に基づく．

- モデル品質と利用時の品質の区別：通常のソフトウェアの品質は製品品質と利用時の品質に分けて捉えられるが[6]，ML システムにおいても類似の捉え方ができる[2][10]．本品質確認フレームワークにおいても，ML モデルそのものの正確性や堅牢性といったモデル品質と，ML モデルを用いて運用を行った際の安定性や有効性といった利用時の品質とを区別して確認を行う．
- 独立した二つの確認視点：品質確認においては，モデル品質と利用時の品質を異なったチームが独立性をもって行う．モデル品質は開発チームが，ML モデルの実現技術やその生成プロセスに関する理解を前提に確認を行う．一方，利用時の品質は運用チームが，それら実現技術等に関する知識を前提とせず，運用の観点からその品質を確認する．
- 差分テストの応用：ML モデルの出力についての正解は未知であるが，運用面からはその出力の値や傾向に関する一定の知識や経験的知見を持っていると考えられるため，それに照らして ML モデルの利用時の品質に関する確認を行う．しかしながら日時の ML モデル生成と運用のサイクルの中で個別詳細な確認は不可能なため，差分テストの考え方を適用して利用時の品質の観点から要確認と思われる個所を抽出する方法をとる．すなわち，直近の過去の予測値との差分や変化率を捉え，それらから要注目点を抽出して，それに基づいた確認を行う．

3.2 全体像

品質確認フレームワークは開発チームによるモデル品質の確認と，運用チームによる利用時の品質の確認との独立したプロセスから構成される．図 2 は全体像を示したものである．

開発チームは一昨日までのデータから特徴量を抽出しそれに基づいて学習を行い，モデル品質の観点から適切に学習されていることを確認してその日に利用する ML モデルを運用チームに渡す．運用チームはそのモデルからの出力と，比較対象となるモデルやデータを用いた出力との差分の抽出を行い，要確認と考えられる点があれば確認を行ってから運用を行う．もしも確認によって問題と考えられるならば，何らかの対応をとることとなる．なお差分の抽出にはそのモデルの利用方法などに依存して多様な方法が考えられるが，今回行った方法については 4 章で説明する．

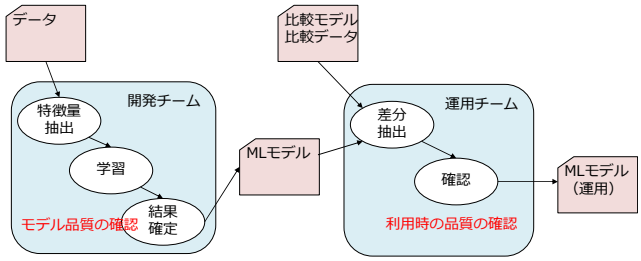

図 2 品質確認フレームワーク

3.3 利用時の品質

ML モデルそのものの品質に様々なものが定義されているが[2][8][10]，利用時の品質については相対的に少ない．それは利用時の品質の多くがその ML モデルを含む ML システムの種類や使われ方，またその対象分野などに強く依存するからと考えられる．従って利用時の品質のカテゴライズは可能であるとしても，実際の品質確認にあたっては，そうした種類や分野を踏まえた具体的な品質の定義が重要となる．

また，本研究が想定する日次の運用においては，利用時の品質の確認は短時間で行う必要があるため，様々な観点から網羅的に確認をすることは現実的ではない．従って，ML モデルの品質の影響を受けやすく，かつ重要性の高い運用上の観点を設定する必要がある．

今回の事例で設定した観点については 4.2 で説明する．

3.4 差分確認

ML モデルあるいは ML モデルを内包する ML システムを利用するという観点からは，内部的な実現技術としての機械学習は一義的には無関係ではあるが，実際にはその影響を受ける．特に，明確なテストオラクルが設定できないといった特性は，品質確認における基準設定を困難とする．

そこで疑似オラクルとなるような比較対象となるモデルやデータを設定し，それらの出力から得られる運用上のデータと，確認対象となる ML モデルの出力から得られる運用上のデータとの差分を比較することで，品質上の要確認個所を抽出するというアプローチをとる．

今回の事例での差分抽出や比較の方法は 4.3 で説明する．

4. 事例

3 章での品質確認フレームワークの有効性や課題について検討するために，事例に対して試験適用を行った．

4.1 金融時系列予測モデル

ここでは金融分野での銘柄の報酬についての予測を行うための ML モデルを事例に検討を行った．

金融分野における機械学習を用いたモデルの構築は，時系列予測に限らず，テキストのセンチメント分析まで多岐に渡る．それらの中には，自然言語処理などコーパスの更新頻度が比較的多くなくモデルの頻繁な更新を想定しないモデルもあるが，今回対象とするモデルは，日次の市場データをモデルに反映するため、更新が頻発することを想定した時系列予測を対象とする．

本研究で扱うモデル[1] は，金融時系列における特徴分析のサンプル数が，画像や文章などの特徴分析と比較して少ないという分析難易度を，クロスセクショナルな予測として従来選択的に用いられているファクターと呼ぶ予測変数により特徴量を拡充することで，これを緩和している．そのため，入力データは，予測する市場の銘柄群を 1 つのデータセットとして，クロスセクションに予測変数を持つ時系列サンプルとして構成される．また，出力結果の設計が先行研究[3] [14] と異なり，サンプル間の優劣の落とし込みとして，パーセンタイルランクにより表現されている点も特徴である．つまり，運用時における出力結果のテストについては，個別銘柄の予測結果に着目しているのではなく，銘柄群の分布を想定して設計されている．

4.2 分位の変動

運用チームによる品質の確認には，出力値における分位の変動を利用する．これは分位の変動がどの程度か，どういう傾向があるのか，ということについて運用チームが一定の知識や経験的知見を持っていると考えられるからである．今回は運用チームとのやりとりを踏まえ，この分位の変動に注目することとした．

4.1 で説明したように，本 ML モデルは銘柄のパーセンタイルランクを出力する．分位とは，各銘柄のパーセンタイル値を昇順に並べ，10 分割したものである．なお上位から第 1 分位～第 10 分位とする．

分位の変動とは、昨日と本日でそれぞれの分位がどのように変わったかであり，その指標に変動数と変動比率を用いる．

- 変動数（以下 gap とも表記する）は、昨日と本日で分位が幾つ分変動したかを表す。例えば、昨日に第 1 分位に含まれていた銘柄が本日では第 3 分位に含まれている場合、2 分位変動しているため変動数は 2 とする。
- 変動比率（以下 rate とも表記する）とは、本日の各分位に含まれている銘柄が昨日も同じ分位に含まれている割合を表す。第 n 分位の場合、以下のように定義される．

$$変動比率 = \frac{昨日も本日も第n分位の銘柄数}{本日の第n分位の銘柄数}$$

出力される個々の銘柄のパーセンタイルランクではなく，上記の指標を用いて分位の変動を捉えることで全体の傾向を把握することを狙った．

4.3 差分の比較方法

本日生成された ML モデル（これを本日モデルと呼ぶ）の確認を行う状況を考えると，4.2 で説明した分位の変動は，昨日の ML モデル（これを昨日モデルと呼ぶ）に一昨日のデータを入力して得られた出力値と，本日生成された ML モデルに昨日のデータを入力して得られた出力値に基づくものである．入力データは市場の変化に伴い日々変わり，またその変化の程度も大きい場合や小さい場合など様々である．従って例えば分位の変動が大きかったとしても，それはデータの変動が大きかったために分位の変動も大きくなったのか，あるいはデータの変動は小さかったのに本日モデルの品質に問題があり大きな分位の変動となったのかを区別することは難しい．

そこで，判断の参考とするために，昨日モデルに昨日のデータを入力して得られる分位を用いる．すなわち，以下の 3 つの分位を比較する．

① 本日分位：本日モデルに昨日データを入力して得られる分位．この分位を本日の運用に利用する．
② 昨日分位：昨日モデルに一昨日データに入力して得られる分位．これは昨日の運用に利用された分位である．
③ 比較分位：昨日モデルに昨日データを入力して得られる分位．比較のために作成する分位

表 1 は上記をまとめたものである．なお，本日モデルの作成には一昨日データが利用されているので，その組み合わせはありえない．

表 1 比較する分位

モデル / 入力	昨日モデル	本日モデル
一昨日データ	②昨日分位	NA
昨日データ	③比較分位	①　本日分位

ここで①と②はモデルも入力データも異なるため，上述したように，例えば②から①への分位の変動が大きかった場合，それがモデルの変化に起因するものか，入力データの変化に起因するものか判別しがたい．しかし，もしも昨日モデルと本日モデルに一定の類似性があると仮定できるなら，②から③の分位の変動と比較することにより，以下の推定ができる．

- ②から①，②から③，いずれでも大きな分位の変動がある場合は，入力データに大きな変動がある．
- ②から①は大きな分位の変動があるが，②～③へは大きな分位の変動がない場合はモデルに問題がある可能性がある

なお分位の変動は，前述した変動数や変動比率の差で比較する方法と，直近 3 日のパーセンタイル値の変化率の差によって比較する方法を試した．ここで変化率は最小二乗法によって求める．

$$変化率 = \frac{\sum_{n=1}^{3}(x_n - \bar{x})(y_i - \bar{y})}{\sum_{n=1}^{3}(x_n - \bar{x})^2}$$

4.4 試験適用

品質確認フレームワークの有効性や課題に関する基本的な初期段階の検討を行うために，本事例に対して試験適用を行った

4.4.1 目的

本試験適用では以下を確認することを目的とする．

- ML モデルが正しく生成されない場合に出力から得られる分位にどのような影響があるか．
- 運用者視点から要確認と考えられるような分位となる場合，それを提案する品質確認フレームワークで捉えることができるか

4.4.2 実験方法

一年間の運用に基づき，本日モデル（以下 model とも表記する），昨日モデル（以下 old とも表記する），劣化モデル(以下 ng とも表記する)を用いて，日次で分位を求めた．ここで劣化モデルは，品質に問題を持つモデルを想定したものである．今回の場合，ある一部の銘柄の予測に問題があり，市場全体の分布を正しく予測できなかったことが，運用期間中に品質が維持できなかったモデルにあたる劣化モデルとして，本日のモデルを作成した際の学習を 70%減らした,学習が不十分なモデルにより再現を試みている．

これにより，本日モデルを正しいモデル（正解）と想定したとき，劣化モデルは正解とは異なる問題含みの分位を生成しうる．実際の状況では正解は未知なので，昨日モデルと劣化モデルから得られる分位の変動から，問題含みの分位を検出できるかどうかを確認する．

なお，変化率を求める際には直近 3 日分の分位を用いるが，一昨日（d-2）と昨日（d-1）については正しいモデルから得られた分位を用いる．すなわち図 3(a)に示すように，想定する状況は昨日までは正しいモデルで運用していたが，ある日（d）のモデルが劣化モデルだったという想定であるため，それに従った計算を行った．

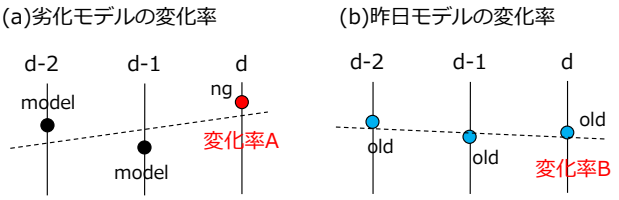

図 3　実験での変化率の計算

上記に基づき，以下の手順で確認を行った．

1. 問題の含みの分位：本日モデルと劣化モデルの出力から得られた分位を比較して，劣化モデルによって得られた分位に問題があると考えられる状況を調査する．
2. 問題の識別可能性：昨日モデルと劣化モデルの出力から得られた分位の変動を比較することにより，上記で識別された問題が発生したことを識別できるかどうか分析する．

4.5 結果

上記の試行結果について述べる．

4.5.1 問題含みの分位

正解と想定する本日モデルから得られた分位と，劣化モデルから得られた分位の変動比率（rate）の概要を図 4 に示す．横軸が時間軸で 10 の分位毎の変動比率を示したものである．なお変動数（gap）についても類似の傾向が観測された．

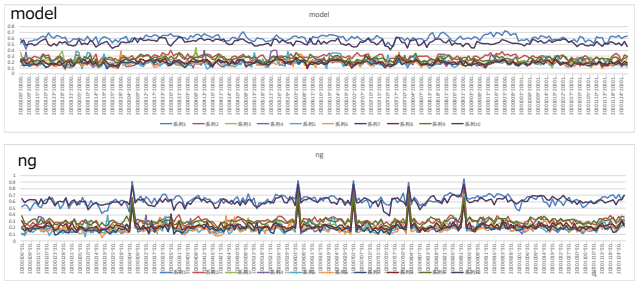

図 4　変動比率（rate）の比較

両者を比較し，要確認の状況，すなわち検出が望まれる状況として以下が確認された．

- 本日モデルにはみられない大きな変化・極端な変化が劣化モデルでみられる．図から直感的に識別できるように，本日モデルでは見られない大きな変動を示す日がある．
- 本日モデルでは一定の変化があるのに劣化モデルでは変化が小さい．詳細データは割愛するが，本日モデルに比べて劣化モデルの変動がかなり少ない日がある．

上記以外にも，本日モデルでは変化が小さいのに劣化モデルでは大きな変化がある状況も考えられるが，今回の実

験では上記の極端な変化を除き，その状況はほとんど観察されなかった．

4.5.2 問題の識別可能性

劣化モデルと昨日モデル分位の変動を比較することで，4.5.1 で識別された状況を検知できるかどうかを分析した．

表 2　変動比率の差（一部）

model	old	ng	model−ng	old−ng
gap=9	gap=9	gap=9	gap=9	gap=9
0.525114		0.648402		
0.575342	0.52968	0.584475	0.009132	0.054795
0.484018	0.525114	0.616438	0.13242	0.091324
0.502283	0.474886	0.634703	0.13242	0.159817
0.511416	0.534247	0.625571	0.114155	0.091324
0.479452	0.520548	0.625571	0.146119	0.105023
0.506849	0.493151	0.557078	0.050228	0.063927
0.547945	0.506849	0.643836	0.09589	0.136986
0.575342	0.52968	0.570776	0.004566	0.041096
0.593607	0.552511	0.579909	0.013699	0.027397
0.488584	0.547945	0.675799	0.187215	0.127854
0.474886	0.506849	0.611872	0.136986	0.105023
0.515982	0.497717	0.534247	0.018265	0.03653
0.479452	0.506849	0.598174	0.118721	0.091324
0.543379	0.515982	0.607306	0.063927	0.091324
0.520548	0.511416	0.593607	0.073059	0.082192
0.543379	0.520548	0.648402	0.105023	0.127854
0.488584	0.534247	0.538813	0.050228	0.004566
0.484018	0.497717	0.607306	0.123288	0.109589
0.474886	0.493151	0.625571	0.150685	0.13242
0.488584	0.461187	0.547945	0.059361	0.086758
0.579909	0.497717	0.611872	0.031963	0.114155
0.497717	0.52968	0.60274	0.105023	0.073059
0.520548	0.502283	0.607306	0.086758	0.105023
0.493151	0.438356	0.474886	0.018265	0.03653
0.56621	0.538813	0.52968	0.03653	0.009132
0.534247	0.570776	0.570776	0.03653	0
0.456621	0.525114	0.584475	0.127854	0.059361
0.484018	0.547945	0.552511	0.068493	0.004566

表 3 変動比率の差（一部）

old−ng

gap=0	gap=1	gap=2	gap=3	gap=4	gap=5	gap=6	gap=7	gap=8	gap=9
0.027273	0.03653	0.182648	0.13242	0.114155	0.027273	0.045662	0.013699	0.004566	0.054795
0.031818	0.018265	0.050228	0.050228	0.041096	0.05	0.018265	0.031963	0.041096	0.091324
0.031818	0.004566	0.004566	0.246575	0.027397	0.036364	0.027397	0.063927	0.063927	0.159817
0.004545	0.004566	0.063927	0.073059	0.009132	0.05	0.086758	0.031963	0.100457	0.091324
0.013636	0.022831	0.059361	0.027397	0.009132	0.090909	0.041096	0	0.050228	0.105023
0.036364	0.018265	0.041096	0.219178	0.018265	0.05	0.027397	0.054795	0.073059	0.063927
0.095455	0	0.063927	0.246575	0.009132	0.018182	0.027397	0.045662	0.086758	0.136986
0.022727	0.018265	0.027397	0.191781	0.004566	0.086364	0.004566	0.054795	0.009132	0.041096
0.009091	0.068493	0.041096	0.059361	0.027397	0.022727	0.013699	0.018265	0.054795	0.027397
0.018182	0.004566	0.059361	0.009132	0.068493	0.018182	0.054795	0.054795	0.059361	0.127854
0.009091	0.127854	0.022831	0.059361	0.027397	0.040909	0.13242	0.054795	0.018265	0.105023
0.054545	0.054795	0.013699	0.031963	0.059361	0.036364	0.03653	0.004566	0.013699	0.03653
0.004545	0.03653	0.050228	0.082192	0.027397	0.081818	0.09589	0.018265	0.018265	0.091324
0.131818	0.031963	0.009132	0.03653	0.004566	0.045455	0.059361	0.045662	0.013699	0.091324
0.059091	0.03653	0.173516	0.041096	0.068493	0.022727	0.018265	0	0.050228	0.082192
0.05	0.027397	0.187215	0.073059	0	0.031818	0.054795	0.031963	0.063927	0.127854
0.131818	0.027397	0	0.050228	0.041096	0.027273	0.050228	0.018265	0.050228	0.004566
0.018182	0.013699	0.027397	0.045662	0.041096	0.072727	0.018265	0.063927	0.045662	0.109589
0.054545	0.086758	0.050228	0.027397	0.018265	0.027273	0.045662	0.009132	0.063927	0.13242
0.063636	0	0.027397	0.013699	0.027397	0.018182	0.059361	0.031963	0.03653	0.086758
0.040909	0.022831	0.009132	0.013699	0.091324	0.063636	0.022831	0.054795	0.105023	0.114155
0.068182	0.077626	0.086758	0.013699	0.196347	0.059091	0.013699	0.018265	0.091324	0.073059
0.045455	0.03653	0.022831	0.004566	0.041096	0.05	0.018265	0.004566	0.100457	0.105023
0.018182	0.022831	0.03653	0.004566	0.159817	0.077273	0.027397	0.073059	0	0.03653
0	0.004566	0.031963	0.068493	0.26484	0.022727	0.004566	0.031963	0.004566	0.009132
0.009091	0.009132	0.018265	0.031963	0.219178	0	0	0.059361	0.03653	0
0.05	0.13242	0.022831	0.073059	0.013699	0.045455	0.009132	0.073059	0.077626	0.059361
0.077273	0.091324	0.004566	0.050228	0.009132	0.004545	0.100457	0.041096	0.031963	0.004566
0.118182	0.013699	0.086758	0.073059	0.013699	0.068182	0.041096	0.063927	0.141553	0.073059
0.15	0.100457	0.013699	0.027397	0.009132	0.05	0.004566	0.027397	0.013699	0.013699
0.172727	0.100457	0.004566	0.018265	0.068493	0.086364	0.077626	0.050228	0.045662	0.03653
0.018182	0.100457	0.073059	0.027397	0.027397	0.036364	0.004566	0.041096	0.027397	0.041096
0.027273	0.018265	0.09589	0.018265	0.013699	0.040909	0.041096	0.013699	0.091324	0.086758
0.159091	0.031963	0.063927	0.004566	0.059361	0	0.018265	0.082192	0.063927	0.164384
0.013636	0.073059	0.118721	0.018265	0.004566	0.040909	0.063927	0.063927	0.03653	0.109589
0.068182	0.068493	0.09589	0.054795	0.041096	0.018182	0.086758	0.041096	0.091324	0.082192
0.004545	0.050228	0.004566	0.073059	0.041096	0.040909	0.009132	0.041096	0.068493	0.050228
0.036364	0.059361	0.03653	0	0.031963	0.009091	0.068493	0.054795	0.118721	0.045662
0.177273	0.114155	0.100457	0.009132	0.050228	0.090909	0	0.073059	0.031963	0.009132
0.336364	0.378995	0.356164	0.47032	0.324201	0.409091	0.3379	0.365297	0.406393	0.315068
0.018182	0.082192	0.018265	0.246575	0.063927	0.040909	0.013699	0.063927	0.054795	0.027397
0	0.018265	0.054795	0.031963	0.004566	0.068182	0.009132	0.018265	0.054795	0.063927

表 2 は第 9 分位における変動比率の比較例である．右から二つめの列が本日モデルと劣化モデルの差(正解との差)，一番右の列が昨日モデルと劣化モデルの差を示す．ここでは差が 0.1 以上のものにハッチングをしている．この二つの列のハッチングが一致していれば，昨日モデルと劣化モデルとの差をみることで正解との差を推定することができると考えられるが，一定の一致はあるものの擬陽性や偽陰性が存在した．

表 3 は劣化モデルと昨日モデルの変動比率の差を示したものである．各行が一日分であり，分位毎に両者の変動比率の差を示している．ハッチングはその差が 0.1 以上ある項目である．大きな変化がある場合には明確にその個所を捉えることができた．表の下部にすべての分位で大きな差が出ている日は，変化率が大きく動いた日と一致する．

一方，本来大きな変化があるのに劣化モデルがそれを示さない状況については一定の把握は可能であったが，表 2 同様に擬陽性や偽陰性が発生するため，その識別は相対的に不明瞭であった．

4.6　考察

今回の試験適用の結果に関して考察を加える．なお本試験適用はあくまで限られた状況を設定したものである．例えば今回は学習が不十分なモデルを劣化モデルとして使ったが，劣化モデルの形態や特性は多様である．従って以下の考察はあくまで今回の試行に限ったものである．そうした理由からあくまで定性的な考察にとどめる．

4.6.1 要確認な状況

要確認な状況を定義することは必ずしも自明ではない．明らかに突出した変化量があればそれを要確認とすることはできるが，そうでない場合，変化量が正解とどの程度の違いであれば要確認とするのかを定めるには，実際の運用への影響などを検討しながら決定する必要がある．

また同じ変動であっても，分位によってその影響は異なりうる．例えば一日で 9 分位近くの変動（つまり最高ランクのものが次の日に最低ランクになる，あるいはその逆）はそれほど多量には存在しないため，逆に変化率は大きくなる傾向がある．分位の変動という指標が，経験的な現象とどのような状況を意味しているかをより精査する必要がある．

4.6.2 品質確認フレームワークの有用性

提案する品質確認フレームワークは一定の有用性はあると考えられる．突出した状況の検知は変動の絶対値だけを見ていてもある程度判断できるが，そうでない状況は昨日モデルとの比較がなければ検知が難しく，差分テストの利用は意味があると考えられる．しかしながら，前項での指摘とあわせ，その閾値や判断基準の設定に関してはさらなる検討が必要である．

一方今回の方法は昨日モデルが本日モデル（正解）とある程度の類似性を示すという仮定に基づいているが，両者はそこまで強い相関を持っているわけではない．例えば変化量について，本日モデルと昨日モデルから得られる変化量の相関係数を求めると，分位によるが 0.3〜0.5 程度の弱

い相関となっている．これはモデルが比較的近傍のデータに大きな影響を受けているためと考えられるが，昨日モデルの利用方法についてもさらに検討が必要と考えられる．

4.6.3 品質確認フレームワークの利用方法

本品質確認フレームワークの利用のユースケースとしては，異常かどうかの白黒をつける判断ではなく，運用の観点からどの程度の品質のモデルなのかを推定する，あるいはその特徴や安定性に関する指標として参考とするという使い方が妥当と考えられる．そういう意味では，今回は分位の変動に注目したが，運用への影響性という観点から他の指標も捉え，多面的な判断を行う方法も考えられる．

5. 議論

以下，関連する議論を行う．

5.1 ML システムの品質保証

ML システムの品質保証については様々な研究や議論がなされており，国内でも品質に関するガイドラインなどが示されている[2][10]．それらでは通常のソフトウェアの品質[6]と同様に製品品質と利用時の品質という品質の区別が反映されているが，製品品質，すなわち ML モデルの実現技術に依存した ML モデル特有の品質に関する議論が多い．利用時の品質は一義的には実現技術に依存するものではなく，システムの外部仕様に依存した利用に関わる品質ではあるが，現実問題として機械学習技術を内包したシステムは，例えば明確な答えのない問題を解くなど，ML モデルの持つ特徴を有しているため，その品質確認には特有の考慮が必要となる．本品質確認フレームワークは特にそうした利用時品質の確認を行うことを意図している．

ML システムの開発や品質確認のフローに関してもオフラインテストとオンラインテストの段階を分けるなど，ひな形的な提示がある[8][15]．本品質確認フレームワークはオフラインテストにおける利用時品質の確認に関わる位置づけとなる．

一方，開発や品質確認におけるソフトウェア工学的な課題についての議論もあるが，対象となる分野，システム，あるいは開発する組織の成熟度などによりその課題は多様であり[2][11]，事例に即した検討が重要であると考えられる．本品質確認フレームワークも日次更新されるモデルを運用するという状況を想定しており，そうした運用の特徴や制約の中での確認方法を検討するものである．

5.2 テスト技法

ML モデルのテスト技法に関しては様々な研究があるが，ML モデルの技術的特徴を踏まえたモデルの製品品質に関わる技法の提案等が多い．それらの中では，差分テストの手法の応用も提案されている[7][9]. 差分テストは類似したソフトウェアは類似した出力を出すという想定の下，疑似オラクルとして利用するものであるが[12]，本品質確認フレームワークでも，利用時の品質の確認に差分テストの考え方を応用している．

一方，品質評価に典型的なルールとしてのルーブリックを活用する手法も提案されている[5]．本稿の試験適用では分位の差分というひとつの観点に注目したが，運用面からの妥当性という観点からの評価にこうした手法を取り入れることも有効と考えられる．

5.3 アジャイルな運用

ソフトウェア開発においてはアジャイルな開発や運用が広まっているが，ML モデルのアジャイル開発や運用に関する議論も行われている．例えば ML システムの継続的デリバリの議論なども行われており，テスト手法やリリース手法を含めた検討が行われている[13]．本研究の想定する日次の更新という状況も，確認だけではなくその先の運用というサイクルを含めたプロセス全体の中で議論されるべきであり，今後そうした視点の拡大も必要になる．

6. おわりに

本稿では日次更新される機械学習モデルの品質確認フレームワークについての提案を行うとともに，金融時系列予測モデルを事例に，初期の試行実験を行った．利用時の品質の確認に関する問題の理解が深まると共に，本品質確認フレームワークの有効性や課題についての知見を得ることができた．今後さらに実用性を高めるための検討を進めていきたい．

参考文献

[1] Abe Masaya and Hideki Nakayama: Deep learning for forecasting stock returns in the cross-section, Pacific-Asia conference on knowledge discovery and data mining, Springer, Cham, 2018.
[2] AI プロダクト品質保証コンソーシアム, AI プロダクト品質保証ガイドライン, 2020.02 版, 2020.
[3] Dennis Olson and Charles Mossman : Neural network forecasts of Canadian stock returns using ac-counting ratios, International Journal of Forecasting, 19(3), pp. 453-465, 2003.
[4] D. Scully, Gary Holt, et al.: Hidden Technical Debt in Machine Learning Systems, NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, December 2015 pp.2503–2511, 2015.
[5] Eric Breck, Shanqing Cai, et al: The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction, Proceedings of IEEE Big Data, 2017.
[6] ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, 2011.
[7] Jianmin Guo, Yu Jiang, et al: Dlfuzz: differential fuzzing testing of deep learning systems. In Proc. FSE, pp.739-743.

ACM, 2018.
[8] Jie M. Zhang, Mark Harman, et al: Machine Learning Testing: Survey, Landscapes and Horizons, IEEE Transactions on Software Engineering PP(99):1-1, 2020.
[9] Kexin Pei, Yinzhi Cao, et al.:Deepxplore: Automated whitebox testing of deep learning systems. In Proceedings of the 26th Symposium on Operating Systems Principles, pp. 1-18. ACM, 2017.
[10] 国立研究開発法人産業技術総合研究所, 危害学習品質マネジメントガイドライン 第 1 班, CPSEC-TR-2020001, 2021.
[11] Lucy Ellen Lwakatare, Aiswarya Raj, et al.: A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation, Proceedings of 20th International Conference, XP 2019, pp.227-243, 2019.
[12] Martin D. Davis and Elaine J. Weyuker: Pseudo-oracles for nontestable programs. In Proceedings of the ACM 81 Conference, ACM 81, pp.254-257, 1981.
[13] https://martinfowler.com/articles/cd4ml.html (2021.2.8 access).
[14] Qing Cao, Karyl B. Leggio. et al.: A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. Computers & Operations Research 32(10), pp.2499–2512, 2005.
[15] Saleema Amershi, Andrew Begel, et al: Software Engineering for Machine Learning: A Case Study, Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, May 2019 pp.291-300, 2019.