KPIツリーを用いた機械学習プロジェクト管理フレームワーク

浜田 伸一郎　吉岡 信和　内平 直志

IT ソリューションへの機械学習適用の期待は高まっている．機械学習を用いる事で，内部動作を全て書き下す事なく，様々な状況に適応した動作をさせる事ができるため，従来では対応不可能だった複雑な業務問題に対応できる事が，様々な分野で実証されてきている．一方，機械学習の動作は開発者にとってブラックボックスとなるため，機能や性能をきめ細かくコントロールすることは出来ない．このことが，開発プロジェクト管理において，従来になかった要求設計・開発計画・進捗管理・品質管理上の問題を引き起こしている．我々は，これらの問題に対処することを目的とした，機械学習プロジェクト管理フレームワークを提案する．当該フレームワークは，KPI ツリーを用いて試行錯誤を含むプロジェクト進捗を管理するとともに，KPI ツリーに含まれる KPI および KGI と対応付く形で，段階的な性能評価を行うことで，評価プロセスの効率性向上とプロジェクトの最終目標と合致した適切な効果測定とを実現する．機械学習を用いた生産設備予知保全プロジェクトに当該フレームワークを適用し，実際の業務での有効性を確認した．

Expectations for applying machine learning to IT solutions are increasing. By using machine learning, it is possible to make operations adapted to various situations without writing down all internal operations. As a result, the capability to deal with complex business problems that could not be handled in the past has been demonstrated in various fields. On the other hand, machine learning behavior is a black box for developers, so it is not possible to finely control functions and performance. That causes unprecedented problems of requirement design, development planning, progress management and quality control in developement project management. We propose a machine learning project management framework aimed at addressing these issues. The framework makes it possible to manage project progress including trial and error, using KPI trees, and it performs step-by-step performance evaluation in correspondence with KPIs and KGIs contained in the KPI tree, which provides efficient evaluation process and appropriate effect measurement in line with the final goals of the project. We applied the framework to predictive maintenance project of production facilities using machine learning, and successfully demonstrated its effectiveness in actual business operations.

1　はじめに

IT ソリューションへの機械学習適用の期待は高まっている．機械学習を用いる事で，内部動作を全て書き下す事なく，様々な状況に適応した動作をさせる事ができるようになる．これにより従来では対応不可能な複雑な問題に対応できる事が，様々な分野で実証されてきている．

一方，機械学習の動作は開発者にとってブラックボックスとなるため，機能や性能をきめ細かくコントロールすることは難しい．このことが，開発プロジェクト推進において従来になかった様々な問題を引き起こしている．以下はその例である．

- 機能実現に対する機械学習のポテンシャルを，非専門家を含むチームでイメージ共有することが難しい．[要求設計]

Machine Learning Project Management Framework using KPI Tree.

Shinichiro Hamada, パナソニック株式会社 デジタル・AI 技術センター, Digital AI Technology Center, Panasonic Corporation.

Nobukazu Yoshioka, 早稲田大学理工学術院総合研究所, Waseda University Research Institute for Science and Engineering.

Naoshi Uchihira, 北陸先端科学技術大学院大学, Japan Advanced Institute of Science and Technology.

- 機械学習を含む機能開発工数を事前に見積もることが難しい．[開発計画]
- 機械学習を含む開発が試行錯誤的となり，プロジェクトの進捗管理が難しい．[進捗管理]
- 機械学習を含む開発結果の機能や性能を保証することが難しい．[品質管理]
- 機械学習の性能評価は良好であっても，実用画面では十分な効果が出ないことがある．[品質管理]

これらの問題を包括的に解決することを目的とした，機械学習プロジェクト管理フレームワークを提案する．また当該フレームワークの実践として，工場生産設備の予知保全ソリューションへの機械学習適用事例を紹介する．

2 機械学習プロジェクト管理フレームワーク

機械学習を用いたソリューション開発プロジェクトは，一般的なシステム開発プロジェクトとは推進方法が大きく異なる．

一般的なシステム開発の場合は，顧客要求に対して要求分析を行い，システムの機能や性能に関する要求仕様を決定する．この要求仕様が開発指針のベースとなり，基本設計・詳細設計・試験仕様などの派生定義を用いて開発プロジェクトを推進管理する．

機械学習プロジェクトの場合は，まずプロジェクト目標に対して，機械学習として解くべき問題を設定する．続いてその問題設定に対して解決方法を検討する．問題設定に対して検討された解決方法がうまく行くかどうかは，解決方法が用いる機械学習手法と利用可能なデータとの相性によるところが大きく，動作検証するまで明らかにならないことが多い．もしうまく行かなければ，同じ問題設定に対して別の解決方法を試行するか，問題設定そのものを見直す．このように，アジャイル型開発プロセスに近い推進方法を取っており，かつ短期間で高頻度にピボットが発生するため，プロジェクトを組織的に推進管理することが難しくなっている．

石川と吉岡 [2] は，機械学習プロジェクトでは事前の見積もりや達成可能な精度の保証が不可能なため，初期フェーズでの要件分析と仕様策定，最終フェーズでの受け入れ検査の策定などが困難であることを指摘している．Vogelsang ら [4] は，インタビューから得られた結果を元に，要求エンジニアリング活動における情報引き出し・分析・仕様策定・検証の 4 つのステップにおいて，機械学習プロジェクトで行うべきことを整理している．

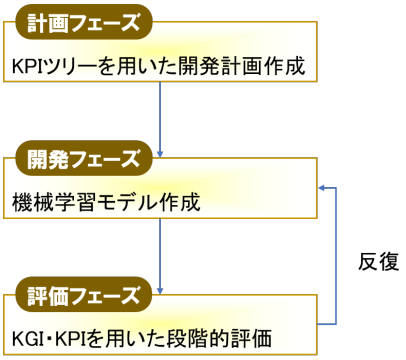

図 1　機械学習プロジェクト管理フレームワークの構成

我々は，機械学習プロジェクトが頻繁にピボットが発生するアジャイル開発に近い進め方になるということを前提とし，機械学習プロジェクトを見通し良くかつ効率的に推進するための管理フレームワークを提案する．当該フレームワークは，3 つのフェーズから構成されている (図 1)．以降の節で，各フェーズの詳細について説明する．

2.1 計画フェーズ

計画フェーズでは，KPI ツリー (図 2) を用いて要件分析と開発計画とを行う．

KPI ツリーとは，解決すべき問題をツリーの形状で論点整理するモデルであり，全体課題の定義およびその達成度指標である KGI(Key Goal Indicator)，部分課題の定義およびその達成度指標 KPI(Key Performance Indicator) から構成される (図 3)．経営戦略策定やマーケティングなどにおいて用いられている．

一般的に，機械学習が解決する問題はプロジェクト全体の一部となる．機械学習による解決結果を，システム機能やシステム利用者と連携させて，プロジェクト全体の最終的な課題を解決する．機械学習による解決が多段階になっていてもよく，その場合，ある機械学習による解決が，その上位の機械学習による解

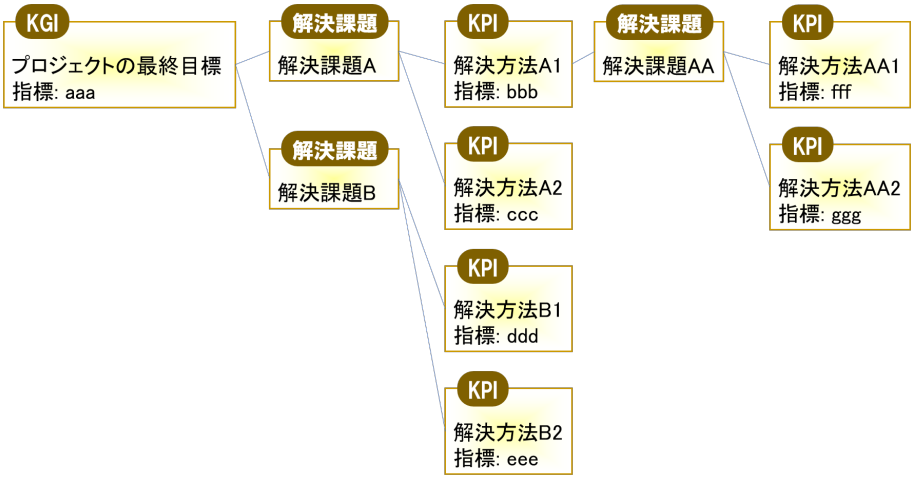

図 2　KPI ツリーの構成

図 3　問題解決構造と KPI・KGI

決の一部として用いられる．機械学習による解決の達成度は KPI で効果測定され，プロジェクトの最終目標は KGI で効果測定される．KPI と KGI はいずれも達成度のための指標であるが，KPI が計測容易性を考慮して設計されているのに対し，KGI はプロジェクトの目標との対応性を考慮して設定されている点が異なる．KPI ツリーは，このような問題解決の依存関係に関する論理構造を可視化する．

本フレームワークの KPI ツリーでは，解決課題と解決方法とを分けて記述する方針を取る．解決課題に対する機械学習による解決方法は一般的に複数考えられるためである．同じ解決課題の下に配置された解決方法群は，同じ解決課題を対象としているため，MECE ではなく取捨選択的である．この点は，経営戦略策定やマーケティングでの KPI ツリーの記法と異なる．

プロジェクトチームは，まず KGI の定義を行う．KGI はプロジェクトの発足目的に対応しているため一意に決定できることが多く，再確認に近い作業になる．この KGI 定義は KPI ツリーのルートノードとして配置する．

次にプロジェクトチームは，KPI を含む解決方針を検討する．解決方針は，プロジェクト目標の中から機械学習用に切り出された部分的な解決課題と，その課題に対する解決方法からなる．このペアをできる限り多数発案し，KPI ツリーのサブノード群としてすべて記述する．ただし前述の通り，解決課題を子ノード，その解決方法を孫ノードに置くものとする．以上により，KPI ツリーが完成する．

最後にプロジェクトチームは，KPI ツリーの子孫ノードとして記述されている各解決方法の採否および優先順位付けを行い，解決優先リストを作成する (図 4)．優先順位の基準としては以下が挙げられる．

- 有用性: 当該 KPI と KGI との間の相関性の高さ
- 実現性: 当該 KPI の期待される達成度の高さ

プロジェクトに納期などの時間制約がある場合は，解決優先リストに実施期間制限を追記する．実施期間を過ぎた場合は，その解決方法を放棄して次優先の解決方法に取り組むこととする．以上により得られた解決優先リストは，次節で説明する開発フェーズにおいて開発計画として用いられる．

KPI ツリーは，次節で説明する開発フェーズの中

	解決方法	KPI	期間
No1	解決方法A2	ccc	2021/8/1-8/31
No2	解決方法A1	bbb	2021/9/1-9/30
No3	解決方法B1	ddd	2021/10/1-10/31

図 4　解決優先リスト

で後から追加してもよい．幅方向の拡大は，代替アプローチを増やすことに相当し，深さ方向の拡大は，解決課題を複数段階に分けることに相当する．

以上で説明した KPI ツリーを用いた推進計画策定方法には，次の利点がある．

機械学習プロジェクトの推進管理　KPI ツリーは，試行錯誤における代替方法を含めた機械学習プロジェクトの推進方針の全体像を示している．プロジェクトメンバーが，作成された KPI ツリーを通じて，プロジェクト推進構造を容易に把握することができる．また KPI ツリーに基づいて，プロジェクトの進捗を管理することができる．

機械学習との相性　機械学習は定量的な指標を用いて最適化を行う技術であるため，本質的に KPI を定義しやすい．その点で KPI ツリーとの相性が良い．

異なる職能を持つメンバーの役割分担　機械学習プロジェクトのチームは，典型的にはドメイン専門家と機械学習技術者とから構成されることが多いが，互いに持っている知識範囲が異なるため，知識の壁を乗り越えて円滑な議論を取ることが難しい場合がある．KPI ツリーを用いることで，役割分担を明確化することができる．例えば，KGI 設計はドメイン専門家，KPI 設計は機械学習技術者が主体で検討するという分担が考えられる．

発想支援　多角的な視点で問題解決を検討することができる．例えば予知保全プロジェクトの場合，典型的な問題設定である設備状態予測に加え，KGI を起点として，保全方法推薦・設備自動補正などの別の問題設定を発想する機会が得られる．

段階的な評価　評価フェーズにおいて，KPI ツリーで示された課題解決の論理構造に沿って，段階的な評価ステップを計画することができる．このことは評価作業の効率化と後戻り最小化をもたらす．詳細は 2.3 節で説明する．

2.2　開発フェーズ

開発フェーズは，計画フェーズで作成した解決優先リストに従って，機械学習モデルを作成するフェーズである．まずモデル作成作業の実行に先立ち，実行上の各種制限事項を確認する必要がある．以下は制限事項の例である．

計算環境の選択　クラウド環境・オンプレミス環境のどちらを利用するのか．クラウド環境が利用できれば，機械学習処理が大規模化しても，計算環境もそれに合わせて強化できるため，実験時間の肥大化を回避できる．しかし，データが社外秘であるため社外に持ち出せない，データが巨大すぎてクラウドへ転送することが難しい，などによりクラウド環境が利用できない場合がある．この場合は，オンプレミス計算環境を用いることになるが，上述のクラウド環境の利点が生かせないため，効率的な開発が困難になる場合がある．

データの入手　データが最初から揃っているかどうか．もし無い場合は，データを収集する手段を解決策に含める必要がある．

Ground Truth の有無　機械学習の目的変数に対応する Ground Truth がデータに含まれているかどうか，特に識別問題の場合は対応するラベルが存在するかどうか．もし無い場合は手動または自動でのラベル付け作業を行うか，教師なし学習または半教師学習などの手法を用いるか，などを検討することが考えられる．

コンセプトドリフトの有無　解決すべき問題が時期などに応じて変化するかどうか．例えば設備の不良発生パタンが経年利用によって変換するなど．もしコンセプトドリフトが起こるのであれば，オンライン学習の仕組みを導入する，コンセプトドリフト発生タイミングを検知する仕組みを導入する，などの対策 [3] を検討することが考えられる．

上記のような実行上の各種制限事項をプロジェクトチームの中で事前確認した後，その確認結果に基づき，必要に応じて，KPI ツリーの修正，解決優先リストの順位変更，実施期間の見直しなどを行う．例えば，ラベル付け作業を行う必要があると判断された場合，KPI ツリーにラベル付け作業をサブ解決問題として追加する．以上により最終決定された KPI ツリー・解決優先リストに従い，機械学習モデル作成を行う．

2.3　評価フェーズ

評価フェーズでは，開発フェーズで作成した機械学習モデルを評価する．開発フェーズと評価フェーズとはループ構造をなしており，評価フェーズでの結果が合格基準を満たさなければ，開発フェーズに戻って機械学習モデルを改善する．このループは評価結果が合格基準を満たすまで続けられる．

評価は最終的に KPI ツリーに定義された KGI を用いて行う必要があるが，KGI による評価は作業工数が大きいため，反復実行が困難であるという問題がある．そこで本フレームワークでは，KPI ツリーの論理構造に沿って，末端ノードから段階的に評価を行う方法を取る．さらに KGI による評価を，現実社会による評価とシミュレーションの評価の 2 段階に分ける．これにより，開発と評価の反復コストを低減する．また評価ステップと評価ステップの間に，モデルパラメタの調整ステップを入れる．これにより，評価の観点の違いによる最適化誤差を吸収することができ，後戻り回数を減少させることができる．具体的なフローは，図 5 の通りである．以降，詳細手順を説明する．

この評価の枠組みは，前節で説明した開発フェーズ，KPI 評価・KGI 簡易評価・KGI 正式評価の 3 つの評価ステップ，および評価間にある 2 つのモデル調整ステップから構成されている．なお解決課題が複数段になっている場合は，KPI 評価が複数ステップになる．以降の説明では，簡単のため，KPI 評価は 1 段として説明する．

まず開発フェーズでは，機械学習手法を元に複数モデルを作成する．最も簡単な方法は，同一の機械学習手法に対して与える超パラメタの組合せを様々変化させて学習する方法である．

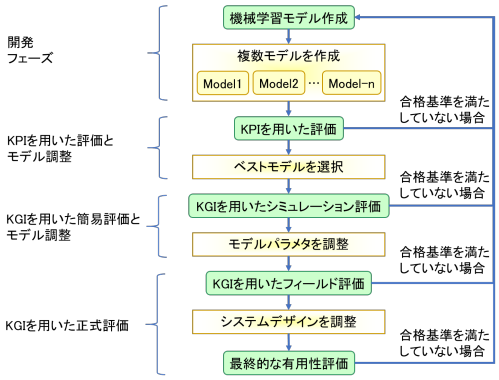

図 5　評価手順のフロー

次に第 1 段階評価として，KPI を用いた評価およびモデル調整を行う．2.1 節で説明したように，KPI は機械学習手法の達成度を示す統計的な指標となっており，コンピュータ上で簡単に計算できる．開発フェーズが用意した複数モデルそれぞれに対して KPI 評価計算処理を行い，評価が最良となった機械学習モデルを選択する．以上の作業により得られた評価結果が合格基準を満たしていない場合，開発フェーズに戻って機械学習モデルを改善する．

次に第 2 段階評価として，KGI を用いた簡易評価とモデル調整を行う．KGI は業務における適用効果を測る指標であるが，業務適用を通じた評価は作業負担が大きい．そのため正式評価に先立ち，コンピュータ上に構築した業務シミュレーションを用いて，業務への適用効果を測定する．この際，機械学習モデルに調整可能なパラメタがあれば，シミュレーション結果が最良となるよう，そのパラメタを調整する．以上の作業により得られた評価結果が合格基準を満たしていない場合，開発フェーズに戻って機械学習モデルを改善する．

最後に第 3 段階評価として，KGI を用いた正式評価を行う．具体的な実際の業務システムに機械学習モデルを用いた仕組みを導入し，現場での業務運用効果を測定する．現場では，機械学習モデル・システム・ユーザが連携して業務を達成することになる．このためユーザインターフェースが評価結果に影響する．

評価結果が向上するようなユーザインターフェースを事前検討する必要がある．効果測定するには比較実験が必要だが，それには A/B テストを行うのが理想的である．しかし現場負担が大きいことが懸念される場合は，機械学習導入後の成績を，過去の生産成績と比較することで代用してもよい．以上の作業により得られた評価結果が合格基準を満たしていない場合，開発フェーズに戻って機械学習モデルを改善する．

以上の評価ステップを全て合格した場合に，KGI による機械学習プロジェクトの有効性の評価結果が得られる．

3 適用事例: 機械学習を用いた生産設備予知保全

2 節で説明した，機械学習プロジェクト管理フレームワークの適用事例の 1 つとして，我々が取り組んだ機械学習を用いた生産設備の予知保全の事例を紹介する．

3.1 予知保全とは

予知保全 (Predictive Maintenance) とは，工場内の設備の不具合や故障をあらかじめ予知し，機械や設備を監視し最適な状態に管理することであり，事前保全の 1 種である．事前保全は，事故が発生してから修理する事後保全に比べて一般に経済的である．

従来は事前保全として予防保全 (Preventive Maintenance) と呼ばれる方法が取られていた．こちらはメーカーや自身の経験から正常に稼働できる使用回数や時間を決め，あらかじめ部品交換などの保全を行うことにより故障を未然に防ぐ方法である．これに対し予知保全は，設備が壊れる予兆や時期などを検出・推定し，そのタイミングで保全することで故障を未然に防ぐ点が異なる．予防保全と比べ，より経済的なタイミングで保全が実施されることが期待される．

近年では，予知保全における設備不良の予兆検出や発生時期推定などの処理に機械学習技術を用いることが試みられている [5] [1]．本節で紹介する事例は，その取り組みの 1 つである．

3.2 計画フェーズ

図 6 は，予知保全を対象とした KPI ツリー作成例である．

工場ラインでの生産において常に意識されるのが金額ベースの費用である．このため KGI は金額ベースで定義されるのが適切である．

機械学習用の問題設定として，設備状態予測・保全方法推薦・設備自動補正の 3 種類を抽出した．このうち設備状態予測については，以下の 4 つの解決方法を検討した．

1. ある未来時刻での不良出力数を推定 (回帰問題)
2. 次にドカ停が到来するまでの時間を推定 (回帰問題)
3. ある未来時刻でのドカ停が発生する確率を推定 (識別問題)
4. 現在からある未来時刻までの間にドカ停が発生している確率を推定 (識別問題)

ドカ停とは不良出力が連続大量発生する設備状態のことである．これらの問題に対する KPI は，回帰問題の場合，RMSE・MAE・MAPE などが候補となり，識別問題の場合，Accuracy や F 値などが候補となる．

各解決方法の優先度は，2.1 節で説明した通り，有用性と実現性の両面から検討される．

有用性は，作成された機械学習モデルの業務上での利用シナリオなどを元に検討する．設備状態予測モデルは，作業員に保全すべきタイミングを発報する機能に利用する．保全方法推薦モデルは，保全作業を行う作業員に保全方法を推薦する機能に利用する．設備自動補正モデルは，回転軸や搬送ベルトのずれなどの状態異常を検出し自己補正する機能に利用する．本事例では，総合的な判断として，設備状態予測モデルが最も有用性が高いと結論付けた．

もう 1 つの観点である実現性については，候補となる機械学習手法の特性，および事前に判明しているデータの特性，などから検討する．設備ログデータは多数の正常時データとごく少数の異常時データからなる．このようにクラスが不均衡な場合は，回帰問題よりも識別問題の方が成績が良いということが経験的に知られている．このことから識別問題の方がやや

図 6　予知保全の KPI ツリー

実現性が高いと考えられる．

以上のレビューにより，設備状態予測問題を識別問題として解決することを第 1 優先として取り組むこととした．

3.3　開発フェーズ

本事例においては，2.2 節で挙げた開発フェーズにおける事前確認項目のうち，計算環境選択・Ground Truth が特に問題となった．

生産設備のログデータは社外秘情報であることが多い．企業の競争力の源泉となる設備設計や生産能力がログデータを通じて外部に読み取られるリスクがあるためである．本事例においても当該リスクありと判断し，クラウド環境は利用しないという結論となった．このため計算環境はオンプレミスのものを利用した．計算資源は限られるため，少量データでの実験からフルデータの実験へと移行するなど，計算量を抑制する工夫が重要となった．

また Ground Truth の問題があった．設備状態予測モデルの目的変数である設備状態ラベルはログデータに含まれていなかった．代わりに，ログデータに含まれる単位時間当たりの製品不良出力数に注目した．

前節で述べたように，設備状態予測問題は識別問題として解くことが有利である．このため製品不良出力数を元に，設備状態のラベル付けを行う方針を取ることとした．しかし稀にしか起こらない設備不良を検出する予兆パタンを機械学習で獲得するためには，大量のラベルデータが必要であることが分かっている．このため，一部データに人手でラベル付けを行い，これを元に自動ラベル付けを行う識別モデルを作成する方針を取った．未来時刻でのドカ停発生予測と異なり，この識別モデルは，過去だけでなく現在および未来の設備状態も判断材料に用いて良いため，本来の問題である設備状態予測よりも，問題の難度は低下している．

3.4　評価フェーズ

ここでは設備状態予測モデルを対象とした評価手順の設計について述べる．

設備状態予測モデルは，生産設備管理システムの中で保全警報を作業員に発報するタイミングを判断するために用いられる．設備保全に関する総合損失は，保全費用と故障損失のトレードオフの上に成り立っている (図 7)．過多な保全を行うと保全費用が膨らみ，過少な保全を行うと故障損失が膨らむことになる．このことから，この設備保全の費用構造の中で，保全警報が効果的なタイミングで発報されているかという

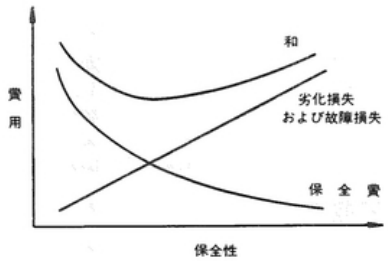

図 7　保全費用に関するトレードオフ [6]

表 1　予知保全の損失項目

保全コスト	部品コスト	交換設備部品の調達費
	稼働コスト	ライン停止による生産機会損失
故障コスト	不良コスト	不良製品に含まれていた材料費
	稼働コスト	チョコ停・ドカ停による生産機会損失

観点で評価しなければならない．

第 1 段階評価では，超パラメタセット 180 種類を元に，180 個の設備状態予測モデルを作成した．これらを F 値で評価し，最も成績の良かったモデルを選択した．

第 2 段階評価では，上述の観点に基づく評価を行うために，設備保全に関する総合損失シミュレーションを用いた．総合損失シミュレーションは，総合損失モデルと作業員行動モデルとを備え，過去の生産設備ログデータを元に総合損失をシミュレーション算出する．総合損失モデルは表 1 のコスト項目から構成される．作業員行動モデルは，以下のようなルールで定義される．

- 発報があれば作業員はすぐさま保全を行う．
- ただし 1 度発報が行われると，一定期間は作業員は発報を無視する．

設備状態予測モデルのパラメタ調整には発報閾値を用いた．設備状態予測モデルは発報に対する確信度を 0 から 1 の実数で出力し，システムはこの確信度を閾値と比較して発報要否を決定する．第 2 段階評価結果が最良となる閾値を選択した．

第 3 段階評価では，工場の生産ラインで稼働する生産設備管理システムに保全警報機能を組み込み，実際の業務における生産効率を測定する．ここで調整するべき項目として発報時のヒューマンインターフェースがある．作業員自体にも保守要否判断に関する知見があるため，機械学習モデルによる予測結果と作業員の知見との両方が相補的に働くヒューマンインターフェースが望ましい．以下の 3 種類の方法を考案し工場関係者を交えて良否を議論した．

- 発報閾値を超えた時にのみ発報を行う
- 常に発報確信度を画面表示する
- 発報確信度を 3 つの値域に分割し，○△×の記号を用いて常に発報確信度を画面表示する

その結果，作業員の保全知識が入れやすく，かつコンピュータの判断を読み取りやすい 3 つ目の方法を採用することとなった．

4　おわりに

機械学習を用いた開発プロジェクトにおける要求設計・開発計画・開発管理・品質管理上の問題を解決する管理フレームワークを提案した．当該フレームワークは，KPI ツリーを用いて試行錯誤を含むプロジェクト進捗を管理するとともに，KPI ツリーに含まれる KPI および KGI と対応付く形で，段階的な性能評価を行うことで，評価プロセスの効率性向上とプロジェクトの最終目標と合致した適切な効果測定とを実現する．また機械学習を用いた生産設備予知保全プロジェクトに当該フレームワークを適用し，実際の業務での有効性を確認した．

当該フレームワークは，予知保全以外の様々な機械学習ソリューションにも適用可能である．今後，様々なソリューションに対して当該フレームワークを適用し，その汎用性を確認していく．

参 考 文 献

[1] Blumberg, A. F. and Mellor, G. L.: Diagnostic and prognostic numerical circulation studies of the South Atlantic Bight, *Journal of Geophysical Research: Oceans*, Vol. 88, No. C8(1983), pp. 4579–4592.

[2] Ishikawa, F. and Yoshioka, N.: How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey, *2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER IP)*,

2019, pp. 2–9.

[3] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G.: Learning under concept drift: A review, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 31, No. 12(2018), pp. 2346–2363.

[4] Vogelsang, A. and Borg, M.: Requirements Engineering for Machine Learning: Perspectives from Data Scientists, *CoRR*, Vol. abs/1908.04674(2019).

[5] Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D.: Deep learning for smart manufacturing: Methods and applications, *Journal of Manufacturing Systems*, Vol. 48(2018), pp. 144–156. Special Issue on Smart Manufacturing.

[6] 平野裕之: ジャスト・イン・生産革命指導マニュアル, 日刊工業新聞社, 1989.