タイトル：GAMP®コンセプトの機械学習への適用

著者：Eric Staib, Tomos Gwyn Williams, PhD, and Siôn Wyn

(Pharmaceutical Engineering, 2023, Vol. 43, No. 1, 14–23)

翻訳：京都大学大学院医学研究科薬剤疫学分野　大学院生　安納崇之（Takayuki ANNO）

この記事では、規制対象のライフサイエンスにおける機械学習（machine learning; ML）のライフサイクル活動について検討する。それは、より広いシステムのライフサイクルの中で ML サブシステムやコンポーネントのライフサイクルと管理を位置付け、文脈化するものである。それはまた、医療用画像認識への ML 応用を実証する事例研究、あるいはプログラム医療機器（software as a medical device; SaMD）[1]によって示される一般的な説明と指針とを提供する。

この記事はより広いシステム、ソリューション、またはアプリケーションの中に組み込まれたＭＬコンポーネントやサブシステムに焦点を当てている。それは、人工知能（artificial intelligence; AI）やML の一般的な入門書や導入書であることを企図したものでもなければ、一般的なコンピューターのバリデーションやライスサイクル活動の導入書でもない。

ML は AI の一分野である。ML システムは、入力（すなわちトレーニング）データから予測モデルを構築し、新しく、これまで見たことのないデータから有用な予測を行うために学習されたモデルを使用する。

ML を利用するほとんどのシステムにとって，従来のコンピュータシステムのライフサイクル、およびコンプライアンスとバリデーションのアプローチの多くの側面は，まだ完全に応用可能である（例えば、ユーザーインターフェース、レポート、セキュリティ，アクセス制御，データの整合性，およびデータのライフサイクル管理の仕様とバリデーションに関連するもの）。

ML コンポーネントという用語の使用は、そのようなコンポーネントが単一なものだと示唆することは意図していない。大抵の場合、ML コンポーネントは通常、入力/データ準備や出力/結果フィルタリングのような多数の機能段階をサポートする「パイプライン」と、共に接続されている一つ以上の中央 ML「エンジン」またはモデルで構成されるサブコンポーネントが複数で構成されている。このような場合、ML サブシステムという用語は最も適切である。著者らは、ML サブシステムと、より広範で包括的なシステム、ソリューション、またはアプリケーションの両方を開発し管理するため、適切なソフトウェアの自動化と他のツールの使用を強く推奨している。また、この記事では、規制によって明確に要求される場合（例えば、デバイスの要件、ユーザーニーズの分析、ヒューマンファクタの評価、臨床試験、および規制当局への提出が考慮される必要があるような SaMD のいくつかの場合）、あるいは、そのような成果物が運用システムの信頼性、保守性、および/または品質、そして意図した用途への適合性にとって明らかに有益である場合を除き、新しい文書での成果

物が必要であるという含意を避けるように努める。(その他の重要な定義に関してはサイドバーを参照)

運用型 ML サブシステムは、その進化に伴い様々な出力を提供するが、そのシステムの検証とバリデーションはこれらの変化に合わせて更新し続けるべきである。これには適切な変更管理、バージョン管理、およびモニタリングを含められていなければならない。加えて、いくつかの ML システムには確率的な要素(ランダムな確率分布やパターンを持つこと)を持ち、それはモデル学習にも関わらず同一の入力に対して結果が異なることを意味する。従って、バリデーションと検証では十分に大きなデータセットと、システム全体の性能において意味をもちかつ代表性があり、そして連続した実行間の小さな出力変動に頑健である性能の要約値を計算することを用いなければならない。

前提要件と背景

ML と多くの従来のアルゴリズムプログラミングとの間にはベストプラクティスにおいて多くの類似点がある。ML の実装を成功させるには、データサイエンティストによる優れたビジネス分析とプロセス理解、効果的な計画、そして優れたソフトウェア開発、エンジニアリング、およびメンテナンスの実践の適用が必要である。適切なデータを最適に選択するには、ビジネスケースと使用目的が完全に理解されなければならず、かつデータ管理は成熟したデータガバナンス戦略によってサポートされなければならない。

性能指標は、あらゆる ML サブシステムの設計において重要である。それらは、どんな出力が生成されるかと、ML の性能を決定づけるための必要な結果、あるいは期待される結果に対してどのようにそれらが評価されるかを定める。これらの指標は、プロジェクト/製造フェーズで説明されているように、全ての ML システムの開発に内在する反復的なトレーニング、評価、そして改善のステップを促すものである。

ML 開発のもう一つの重要な側面は、開発プロセスへのデータとメタデータの密な統合である。データ中心の開発という用語がこの事を反映して時々使用される。その結果として、データの取得、選択、分類、クレンジング、拡大に対する管理など、データは細心の注意を払って管理されるべきである。

他のソフトウェア開発と同様に、ML 開発にはそのシステムの複雑さと新規性に見合ったビジネスリスク、技術的リスク、およびプロジェクトリスクを含む活動がある。これらのリスクを管理することは、問題を認識し、緩和段階や修正段階をとるか、あるいはプロジェクトを終了するかどうかを判断するための全ての開発段階で、優れたプロセス/ビジネス分析、リスク分析、コスト/ベネフィット分析が必要になる。

開発計画では人的要因やバイアス、プライバシー、セキュリティ、そして法的責任を考慮する必要がある。これには透明性と、結果の再現能力、結果の適切な解釈能力、モデルの適用方法に対する適用可能性を理解する能力への理解を要する。

リスクのレベルは使用目的による。バリデーションと管理の程度、厳密さ、そして文書化

には人の関与レベル、(治療や診断、臨床管理の推進、または臨床管理の情報提供のための)医療上の決定に対しての情報の重要性、そして医療の状況や状態（重篤な、深刻な、あるいは深刻ではない）などの要因を考慮すべきである。

ISPE GAMP® Records and Data Integrity Good Practice Guide: Data Integrity by Design [2], 付録 S1: 人工知能: 機械学習では、ML フレームワーク内のデータのライフサイクルを特定し、GAMP データライフサイクルと GAMP システムライフサイクルの両者との関連を強調する。より広範なデータ整合性（data integrity; DI）のトピックもこのガイドで論じられている。

ML サブシステムのデータサイクルの概要

以下は、ML サブシステムのライフサイクルモデルの概要である（図 1 参照)。構想、プロジェクト/製造、および運用を含む、GAMP5 全体のシステムライフサイクルと一致するフェーズ用語が使用されている。次に、SaMD 製品の具体的なライフサイクル活動を紹介した事例研究を示す。GAMP® Good Practice Guide: A RiskBased Approach to Regulated Mobile Applications[1]との整合性を保つため、フェーズ用語にはプロジェクトと製造を含んでいる。

構想フェーズでは、ビジネスのニーズや機会は、特定され、明確化され、そして合意に至る。解決すべき具体的な問題が定められる。初期データが識別され（データウェアハウスやデータレイクからの可能性もある)、選択され、「症例データ」として準備される。プロトタイピングにより、適切なアルゴリズムやハイパーパラメータの評価と選択、および学習プロセスを制御するために用いられる予備的なハイパーパラメータ値が可能となる。例えば、隠れユニット数などのネットワーク構造を決定する変数や、学習率などのネットワークの学習方法を決定する変数がある。データ管理は、最初にケースデータが収集されるこのフェーズから開始になる。

プロジェクト/製造フェーズでは、定められた計画に従って、選択されたテクノロジーと技術アーキテクチャが定められる。プロジェクトベースの構成と変更管理など、正式なリスク管理活動が他の支援活動と同様に開始になる。ML サブシステムにとってのプロジェクト/製造フェーズ活動は典型的には直線的よりはむしろ反復的であり段階的でもある。これらの反復活動はモデルの設計/選択、エンジニアリング、モデル学習、テスト、評価、およびハイパーパラメータのチューニングを含んでいる。

データ管理は、新しいデータ取得、安全な保管と取扱い、準備（ラベリングを含む)、トレーニングとバリデーションデータセットへのデータの分割など、もう一つの重要なプロジェクト/製造フェーズ活動がある。モデル開発段階では、トレーニングデータセットはモデルのトレーニングのために使用され、バリデーションデータセットはモデルのハイパーパラメータを調整しながらモデルの偏りのない評価を提供するために使用される。交差検証実験などのある特定のシナリオでは、特定のデータセットトレーニングとバリデーション役目を果たすことがあるが、実験の同じ反復の中ではそうではない。テストデータはすべてのトレーニングとハイパーパラメータのチューニング活動から除外され、代わりに、包括

的なシステム内の最終的なモデルの偏りのない評価を提供するために使用される。通常、ML コンポーネントのより広いコンピューター化されたシステムへの統合とターゲットあるいはテストデータセットを使用して、受け入れ活動とリリース活動が実行されるような他の環境への展開がある。

運用フェーズでは、システム性能の監視と評価が行われる。新しく（生の）データが利用可能になると、さらなる機器構成／コーディング、チューニング、トレーニング、テスト、そして評価が実行される。新しいデータの利用可能性と継続的な性能評価と品質チェックは、事前事後の両者で、性能向上や使用範囲の変更の機会をもたらすため、生産活動と運用活動を交互に行う緊密で反復するループになる可能性がある。このため、効果的な変更とコード、データ、モデルなどの ML システムの全構成要素へ適用された機器構成管理が必要となる。

ML サブシステムのライフサイクルフェーズ

以下のセクションは、ML サブシステムのライフサイクルでの典型的な活動の説明ならびに考察を行い、記事の最後に実例となるケーススタディーの例 [3] によりサポートされている。

構想フェーズ

このフェーズの目的は、ML サブシステムの予想される開発コストと運用上の利点についての洞察を提供することである。ここでは、なぜ ML ソリューションを組み込む必要があるかについての判断または根拠を含めたいと思う。このフェーズは、コスト、開発リスク、期待される性能に基づいて、どの ML アルゴリズムが開発を検討されるべきかを調査し、研究する機会も提供する。また、このフェーズでは、初期のケースデータの収集やそのデータの特性を理解するための取り組みも行う。

ビジネスニーズと機会の特定

ビジネスニーズが開拓、分析され、全体的なプロセスとワークフローが定められ、かつ合意に至り、提案されたアプリケーションがそのプロセスをどのようにサポートするかが特定される。この分析はデータの利用可能性、展開用ハードウェア、法的責任、規制および知的財産（intellectual property; IP）の要因などの制約を決定づけるのに役立つ。また、ソース、構造、形式、セグメンテーションなどの詳細なデータ関連要因も考慮される必要がある。

問題の定義

この段階では、初期の要件が仕様書に含められることがある。この初期の「要求仕様書」は開発を推進しシステムと ML サブシステムに必要な機能を定義する。

統合や展開の制約など非機能要件も、ML アルゴリズム選択の情報を与えるために、この

初期段階で考慮されるべきである。非機能要件は性能指標の初期セットを含む。これらは、ML サブシステムの出力ならびにこれらの出力が定義された期待値とどのように比較されるかについての詳細な説明である。この比較により、サブシステムの性能の定量的な測定がもたらされる。これらの測定値は ML サブシステムモデルのトレーニング、評価、チューニングを推進する。その性能指標は、開発、トレーニング、再トレーニング中に変更される可能性がある。他の非機能性要件としては、ハードウェアの選択などの展開の制約や、速度や容量などの性能の制約がある。

プロトタイピング

ML プロジェクトは、他のアプリケーションや使用ケースのために開発され、適用されたアルゴリズムや技術を展開することで大きな利益を得る事ができる。この段階の目的は、どのアルゴリズムとリソースが最もプロジェクトを成功に導くであろうかを特定するために、調査と初期プロトタイピングを行うことである。

ML の分野では、選択できるアルゴリズムとモデルアーキテクチャは様々で、ますます広範になっており、各アルゴリズム内には多数の調整すべきハイパーパラメータがある。新しいシステムにとって、アルゴリズムの選択があまりにも明確であり、この段階でコンポーネントを完全に指定して開発に進む決定ができることの可能性は少ない。どのアルゴリズムが最も適しており、どのようにトレーニングされ評価されるかを決定するためには、候補となるものは運用、性能、そしてもし関連する場合には規制要件に対して評価される必要がある。これらの活動は、相応しいモデルの予測性能の早期指示とそのシステムが性能レベルを達成する可能性を提供する。

データ取得と選択

データの初期セットは、プロトタイピングの開始点を提供するため、既存のビジネス活動から収集されるか、あるいは収集する必要がある。ひとたびそれが特定されると、この段階では、モデルのトレーニングと評価のためのデータを準備するために必要なことを決定する。これにはフォーマット、クリーニング、特徴抽出（データ変換と総称する）を含む。また、プロトタイプのサブシステムが評価されるトレーニング入力を提供するために、データがラベリングされる必要が生じる場合もある。この段階では、データが完全であることは期待されていない。なぜなら、後続の段階で、追加のデータが必要であり、そのデータを取得およびラベリングする計画があるかどうかで特定するからである。しかしながら、将来の評価を損なうことがないためにもケースデータをトレーニングセットとバリデーションセットへ分割することが重要である。トレーニングデータには、偏った人間の判断が含まれていたり、不平等が反映されていたり、あるいはトレーニングデータでグループやクラスが過大あるいは過小に示されている欠陥のあるデータサンプリングによってバイアスが生み出されている可能性がある。そのようなバイアスのリスクを制御するために、適切な手段を適用

する必要がある。

プロジェクト/製造フェーズ

このフェーズからの成果は、広範な性能評価手段とともに包括的な IT システムに統合された ML サブシステムの実装である。これに不可欠なのは、モデルのトレーニング、チューニング、および評価をサポートするトレーニングおよび性能評価のインフラストラクチャーの開発である。モデルの構築やデータ準備をサポートするツールもまたこのフェーズで開発されることもある（トレーニングデータのラベリングをサポートするツール等)。

このフェーズは、連続した ML サブシステムのバージョンが指定、設計/選定、実装、トレーニング、チューニング、評価される反復的なアプローチに従う。このフェーズは、性能を最適化するために、設計、実装、サブシステムのハイパーパラメータの選択を繰り返し改善する一連の実験で構成されている。

プロジェクトデータ管理

プロジェクト/製造フェーズの開始前に、ケースデータがプロジェクトライフサイクルの要件を満たしているかどうかを判断する必要がある。例えば、モデルをトレーニングするための十分な量のデータがあり、予測されるリアルワールドデータを網羅するだけのデータ範囲を備えていることがある。もしそうでない場合には、追加データが必要とされ、それは別の取得プロジェクトが必要となる場合がある。このフェーズもまた使用目的に対してのデータの適切性を判断し、サブシステム開発のための準備を行う。活動にはフォーマット仕様、選択、データの注釈とクリーンアップのためのアプリケーションツールが含まれる。

データに必要な範囲と形式は、以前に取得した性能指標によって決定される。例えば、画像解析の物体検出のタスクでは、性能指標はグランドトゥルースと AI によって予測された結果との一致として規定される。グランドトゥルースとは、モデル/分類子の出力が最終的に評価と比較をされる際の承認された外部基準として機能する一式の結果である。これを達成するには、グランドトゥルースデータと AI の出力が、されるべき測定を可能にする比較可能な形式であらなければならない（例えば、画像分割による)。分類タスクの場合、特定の特徴を含んだものとしての画像の単純なラベリングで十分な場合がある。

モデル要求仕様

この段階は、プロジェクト/製造フェーズにおける「トールゲート」とみなすことがあり、ここでは前のフェーズから得られた情報が文書化され、プロジェクト/製造フェーズのための情報に基づいた詳細な計画とともに提示される。その目的は、続行するかどうかの決定をするために、ML サブシステムの予想されるコスト、リスク、および利益に関する情報を提供する事である。この段階で提示される情報は、データ取得、管理、および開発に必要とされる追加投資がビジネスニーズを実現するという確信を与える。

構想フェーズで得られた情報と経験は、プロジェクト/製造フェーズの開始時に、ML サブシステムを可能な限り確実に指定および設計し、リスクの見積もりを含むその配信の計画作業を可能にするために利用される。この段階での活動は、主要なコンポーネントを特定し、どのように統合して解析を実行するかによってサブシステムの初期設計を策定することが含まれる。設計の決定は、前のフェーズのプロトタイプソリューション開発で得た実際の経験に大きく依存する。さらに、サブシステムの仕様が作成され、それにはサブシステムの入出力データの形式と性能指標の定義が含まれる。

プランニングでは、タイムラインや関連リスクの見積もりとともに、開発取り組みの詳細な内訳を含む。プロジェクトのリスク分析はこの段階で実行されることができ、最も失敗しそうな項目を判断し、リスクを軽減するための適切な軽減措置や代替ソリューションを提供する。プランニングには ML サブシステムの開発環境の仕様も含まれ、それはソフトウェアライセンス、コンピューティングリソースおよびストレージリソースという形で、独自の予算上の意味合いを持つ。開発運用とハードウェアインフラストラクチャーは、ML コンポーネントのトレーニングと評価をサポートするように設定されている。これらには、ローカルとクラウドベースの計算の任意な組み合わせを適用する、コードとデータのバージョン管理がされたリポジトリを含むことがある。このフェーズは研究に焦点をあてた言語とプラットフォームを使用する場合があるが、その後の技術的または IP 侵害の問題がないことを保証するためにも最終の展開要件とプラットフォームも考慮する必要がある。

モデル設計と選択

ML モデルのベースライン アーキテクチャはこの段階で選択され、設計される。プロトタイピングフェーズから得られた知識は、(性能などの) 機能的、非機能的の両者において、モデル要件を最も満たすことができるものとして特定された単一または少数の候補アルゴリズムを特定するため、ここで適用される。異なる ML アルゴリズムクラスにまたがるモデル選択を可能にするため、要件は十分に広くすることができる。それぞれのアルゴリズムを最適化するために必要な労力が大きくなる可能性があるため、データサイエンティストはこの段階であまり多くの候補アルゴリズムを選び過ぎないように注意する必要がある。もし、候補アルゴリズムの数が 3 つ以上になる場合には、サイエンティストは過度な最適化を回避するため、プロトタイプ段階まで戻りいくつかのアルゴリズムを削除することを望むことがある。

基礎となる ML アルゴリズムの選択が、各モデルの一連のハイパーパラメータを導く。その後の開発プロセスの反復が、モデルのテスト結果によって作動するアーキテクチャを改良する。

モデル/データ エンジニアリング

この段階は、モデルアーキテクチャと、モデルのトレーニングとハイパーパラメータのチ

ューニングを可能にするデータ入力と評価のための周辺インフラストラクチャーの構築を担っている。タスクには、トレーニング反復のためのデータを選択、準備、管理、維持することと、異なるハイパーパラメータの試行とアーキテクチャの異なるバージョン間での比較を可能とする結果の記録が含まれる。ひとたびセットアップされると、それからインフラストラクチャーはモデルのハイパーパラメータを修正する一連の試行を実行するために用いられ、最良のモデル性能をもたらすハイパーパラメータセットが決定される。

モデルトレーニングとハイパーパラメータの最適化

この段階では、様々なハイパーパラメータ値（例; 隠れユニット数や学習率）で一連のモデルインスタンスをトレーニングすること、ならびに結果を記録することを担っている。ハイパーパラメータの最適化には、手動による選択と各反復後のパラメータの変更、あるいは徹底的な検索やハイパーパラメータ空間のより効率的なベイジアン最適化を利用した自動化プロセスを要する場合がある。

ほとんどの ML アルゴリズムは多くのハイパーパラメータを持ち、それゆえに最適化するための大きなハイパーパラメータ空間を定義する。しかしながら、プロトタイプフェーズで得たアルゴリズムと問題領域の知識を応用することで、データサイエンティストが事前に決定し固定できるハイパーパラメータの値のサブセットを特定することが出来るようになり、従って、パラメータ空間を大幅に削減することができる。自動化されたハイパーパラメータのチューニングを可能とするライブラリやインフラストラクチャーは存在するが、データサイエンティストがハイパーパラメータのチューニングに完全に他人任せの方法をとらないことをお勧めする。各実験の実行を最適化するためにハイパーパラメータのサブセットのみを許可することで、ハイパーパラメータ検索空間の実験をより小さな分画へ分ける事は、モデルトレーニングとモデル性能に関するハイパーパラメータが持つ影響についての有益な知見を得る事ができ、さらに効率的なチューニング段階へと導くことになる。

この段階からの出力は全てのトレーニングデータと最適あるいは最適に近いハイパーパラメータのセットを用いてトレーニングされたモデルである。これは既存の固定アーキテクチャとバリデーションデータを使用して評価されたパラメータが与えられた最適なモデルと考えられる。モデル設計→モデルエンジニアリング→ハイパーパラメータチューニング→モデルトレーニング→モデル評価の反復により、最新モデルと前回のモデルの性能に関する洞察が明らかになる。これにより、性能を改善するためにはモデルアーキテクチャとトレーニング選択がどのように変更されたらよいかに関するさらなるエビデンスが得られ、それから再設計あるいは代替モデルの選択が実行され評価されることになる。

評価とモデルテスト

ここでは、前回のトレーニングと選択の反復より最適な性能モデルがバリデーションデータにさらされる段階になる。前回の反復のトレーニングから除外されたバリデーションデ

ータがモデルに渡され、モデルの性能が評価される。公正な比較にとっての重要な要件は、各候補モデルに同一のトレーニングとバリデーションデータのセットを適用することである。その結果はゴールドスタンダードのラベリングと比較され、集計ならびに指標とする性能指標、あるいはスコアカードを作成し、現在の性能についての情報を提供し、必要とされる場合に後に続く反復を推進する。

多くの ML ライブラリは、バリデーションデータをトレーニング機能に組み入れ、このプロセスの多くを自動化する。しかしながら、データサイエンティストはモデル評価に対し定量的な指標に頼ることには注意を要する。バリデーション結果の視覚的な定性評価は、モデルがどのように動作するかについてのより良い洞察をしばしば導き、共通のエラーモデルが特定され、対処されることができ、必要な出力とのより良い整合性を提供するために性能指標を決定的に改良することが可能になる。このため、ハイパーパラメータのチューニングの時には、多くの開発環境によって提供された完全に最適化されたハイパーパラメータのチューニング機能のみに依存するのではなく、専門家とドメインの知識を利用することが望ましい。実際には、ここでは一連のチューニングの実験で構成されるハイブリッドアプローチが含まれ、それは性能指標によってハイパーパラメータのサブセットがチューニングされ、結果の手作業による解釈と定性分析が交互に行われ、次のチューニング実験のセットを決定するかチューニング活動を終了とする。

リリース前製品の性能評価と総合的な性能指標についての詳細な説明とエビデンスは、データサイエンスに基づいた期待値である。

目標モデルの性能が達成され、かつ/あるいはアーキテクチャのそれ以上の変更が特定されない時には、最も性能の良い ML モデルが包括的な IT システムに統合される候補として選択され、展開される。この選択は、バリデーションデータセットでの性能の非機能要件だけでなく、アルゴリズムの管理のしやすさ、目標となる展開環境での展開のしやすさ、他にはランタイムなどの非機能要件など、要件に定義された基準に基づいている。

モデルの統合と展開

この段階で、ML アルゴリズムとモデルは、迅速なプロトタイピングと実験をサポートする開発環境コードから、より効率的で展開環境と長期的な維持管理へより適した展開ターゲットコードに移行される。このプロセスは、候補アルゴリズムのプロトタイピングと実験をサポートするために設計されたコードの多くを削除することが含まれる。これには、採用すべきパラメータとアルゴリズムの選択肢を特定し、望ましい特性や性能を得られなかった候補アルゴリズムを削除することを含んでいる。

このフェーズの鍵は、コードの推論モジュールを残りのコードから分離するモジュール化である。推論モジュールとは、ML サブシステムの出力への入力としてのテストデータやまだ見たことのないデータの順向性のパスに関連したコードの構成要素である。推論とは、生データを入力として受け取り、出力を提供するコードのモジュールであるサブシステム

の順向性パス実行を指す。ここでは、グラウンドトゥルースに対する出力のバリデーションに関する機能、あるいはモデルパラメータやハイパーパラメータの変更に関連したコードは除外する。

ML アルゴリズムは通常、トレーニング、実験、およびハイパーパラメータのチューニングをサポートするために調整された開発環境で開発される。これらの環境は、必ずしも展開要件と一致するとは限らない。その場合、適切なコードレビュー、検証、テストと共に、コードとトレーニング済み ML モデルをランタイム環境に移植する必要がある。もし必要であるなら、ランタイム環境への移植に要する最小限のコードは推論部分になる。統合には、ML サブシステムと包括的な IT システムとの間のインターフェースの仕様と実装も必要である。

推論と同様に、パイプラインの性能評価のコンポーネントはトレーニングコードベースに存在する。しかし、これは完全なパイプラインの性能評価として実装される必要があり、より適切な開発および/またはランタイム環境に移植する可能性がある。

他のソフトウェアシステム開発と同様に、ML 開発にもシステムの複雑さや新規性に見合ったビジネスリスク、技術リスク、プロジェクトリスク活動がある。

受入とリリース

ML サブシステムのリリース、維持管理、性能検証のための最終的なインフラストラクチャーは、このフェーズで開発される。サブシステムの開発、リリース、維持管理に関するプロセスは、開発中の ML アルゴリズムの機能を検証するかどうか、また、いつどのように検証するかを規定する。ML モデルのトレーニングや場合によってはチューニングをこのプロセスの中に含めるかどうかに関しては選択される必要がある。例えば、コードの機能を検証するために、定期的にテストデータ上で完全なモデルのトレーニング、ハイパーパラメータのチューニング、モデルの性能を実行することが決定される場合がある。あるいは、モデルのトレーニングとハイパーパラメータのチューニングは、コアコードまたはインフラストラクチャーの一部ではなく、検証プロセスから除外されているとみなす場合もある。少なくとも、プロセスは、ML サブシステムの検証がどのように行われるかを規定し、適切に文書化する必要がある。このようなプロセスの実行が、ML サブシステムの最初のバージョンのリリースにつながるはずである。

運用フェーズ

このフェーズでは、ML サブシステムが継続的に監視およびメンテナンスされる。これには、結果が事前に設定された制限値から逸脱する場合、手動による監視を含む場合、または

その組み合わせによっては、アラートを出すための自動化を含む場合がある。性能の監視による結果として、サブシステムに影響を与える変更が必要になる場合がある。このような場合、メンテナンスと性能評価プロセスは、代替 ML モデルの再トレーニングと採用をサポートするために強固で十分である必要がある。このような変更は、組織の変更管理プロセスを遵守し、現在および将来の精算データへの変更の影響を考慮したリスクベースの評価を活用して行われなければならない。

典型的な要求は、システムが入力データの特定のサブクラスに一般化することが苦手ということである。典型的な解決策は、このサブクラスのデータを取得してトレーニングデータセットに統合することである。追加トレーニングデータの統合は、性能の全ての変化が測定され、検証され、理解されるという点で体系的でなければならない。例えば、追加データを取得した時点で、その一部をトレーニングセットに割り当て、残りは全てのモデルトレーニング活動から除外することができる。モデルのトレーニングは、データのサブクラスが追加されると、さらに性能を試されるようなデータが含まれるため、性能が全体的に低下する結果となる可能性があることを認識し、拡張されたトレーニングデータセットを用いて進められる。ひとたびトレーニング、テスト、チューニングが完了したら、改訂モデルの性能は、課題が増えたため性能が低下する可能性があることを想定のもとに、拡張されたテストデータセット上で最初に元のモデルによる評価プロセスを実行することで、ステージングする必要がある。そして、改訂モデルによる評価プロセスの実行は、性能指標が望ましい許容レベルを達成することの期待をもとに行われる。

この例から分かるように、システムと ML サブシステムの運用とメンテナンスはそれ自体が元の開発作業のトレーニング、テスト、チューニングのサイクルに従う反復プロセスであり、定義されたデータガバナンスと継続的な性能評価を通じて新しいデータの適切な管理を伴うものである。

事例研究

この事例研究は、歯科咬翼法 X 線のチェアサイドでの分析のためのアプリケーション開発について説明するものである。咬翼法 X 線は、通常、口の左右の歯根を含む上下両方の歯を示す。それらは、歯周病や歯間の齲歯などの複数の疾患の診断と経過観察のための補助手段として使用される。咬翼法 X 線は、口の中の歯と舌の間にセンサーを入れ、口の外側から X 線源を当てて撮影する。その後、そのセンサーを取り外し、デジタルスキャンが行われ画像が提供される。X 線検査は、臨床検査だけで検出される齲歯病変よりも、検出される齲歯病変の数を増やすことができる。この使用は、英国保健省の FGDP（英国）ガイドライン文書[4]で推奨されている。それにも関わらず、システマティックレビューでは、歯科医師による X 線での脱灰の検出の診断感度は 37%しかないとの低さが一貫して報告されている[5]。

「この製品の目的は、齲歯として臨床的に知られている虫歯の早期段階を検出することで

す。早期の齲歯は、咬翼法 X 線で歯の外側のエナメル質表面の外観の微妙な変化によって示されます。これらの小さな変化を検出するのは難しく、特に歯科診療所での照明条件が悪く時間的なプレッシャーが存在する中では困難です。初期段階の齲歯を見つけられないことは、歯間クリーニングやレジン浸潤法などの予防治療を行う機会を逸することになり、さらなる虫歯の進行やドリルや浸潤法などの修復治療の必要性につながる可能性があります」[6]。

本製品は、一連のアルゴリズムを展開して早期虫歯の咬翼法 X 線を分析し、AI が早期の齲歯の徴候となる画像バイオマーカーを検出した領域を矢印で示し、歯科医が詳しく観察する価値ある領域を強調表示する。グラフィカルユーザーインターフェースの操作により、歯科医が矢印を移動、削除、または追加ができる[3]。

本製品は、スタンドアローンのアプリケーションとして、歯科医の既存の画像管理ソフトウェアへの統合されるものとして、あるいはウェブホスティングの分析サービスとして、複数の形態で提供される。EU 医療機器指令[7]のもとでは、エナメル質のみの早期齲歯の補助診断のために、資格を有する歯科医が使用する安全性クラス 1 のプログラム医療機器として登録されている。この製品は ISO13485 規格に従い、開発とリリースがされている。

早期齲歯の検知器のビジネスチャンスと健康上の利点は、最小限の介入あるいは最小限の侵襲的な歯科医療にある。これは、従来のドリルや充填治療の使用に先手を打ち、それらの使用を最小限に抑えるため、早期予防処置として好まれる歯科において先駆的なアプローチである。したがって、ドリルが必要になるまで齲歯または虫歯が歯の深部に入り込むまで待つのではなく、虫歯がエナメル質表面の外側に限定されている早期の時点で病気を発見し、高フッ素の歯磨き粉や歯科衛生士への受診などの非侵襲的な治療で治すことができる[8]。

本製品の機能要件は、より適切な情報に基づいた診断や治療方針について判断するための、虫歯の根拠を強調する支援ツールを説明するために策定された。診断補助機能は、歯科医の既存の作業経路に適合するように選ばれた。また、常に診断や治療方針の最終決定を下すトレーニングを受けた臨床医のみが本製品を使用できるという点で、補助的な性質は規制安全クラスの意味合いを持っていた。本製品の重要な要件は、臨床医がより十分な情報に基づいた決定が下せるように明確な指標を提供するが、臨床医の行動を邪魔したり、そうでなければ干渉したり、置き換えたりしようとはしないことであった。

また、本製品は歯科医の臨床ワークフローにシームレスに組み込まれることが求められた。咬翼 X 線の必要性がひとたび特定されると、口の両側一つずつ一対の咬翼 X 線を取得し、チェアサイドのコンピューターで X 線を直ちに分析し、すぐに患者へ選択された治療方針を知らせるワークフローをとっている。注：通常、歯科医は各咬翼法 X 線の分析するために短い時間しか持たず、その間に早期齲歯に加えて様々な状態を調べている。そのためには、咬翼法 X 線で早期齲歯だとする領域を強調する完全に自動化された分析が必要であった。

非機能要件としては、通常は専門的なハードウェアやインターネット接続に依存することのない PC であるチェアサイドのコンピューターで動作することが挙げられた。
また、分析にかかる時間は高速で、歯科医の現在の画像分析と臨床報告にかかる時間である 20 秒を増大させない必要があった。

齲歯検出性能に関して、これまでの研究では一般診療所の歯科医が早期齲歯の約 40%を検知していることが示されている。性能目標は、誤検出（すなわち、偽陽性）を許容できないほど高めることなく検出率を上げることであった。誤検出は望ましくはないが、治療方針が非侵襲的であるので、それは患者にとって害を与えることにはならない。

歯科咬翼法X線の画像分析に関連した技術について、実装と性能のエビデンスに着目し、関連する文献と出版物を網羅的に検索した。その結果、深層学習アルゴリズムの歯科画像の解析への適用に関する情報は得られたが、公表された咬翼法 X 線に関するケースデータはなかった。重要な公開データセットはないことから、プロジェクトの一環としてデータ取得を実施する必要があり、したがって、使用される ML アルゴリズムは、適切な精度による予測モデルを生成するために大きなサンプルサイズを必要とすべきではないという要件を含むものであった。

隣接面の齲歯の有病率が高い一つの現場（すなわち、歯科医院/診療所）から収集された 130 の咬翼法 X 線からなるデータの初期セットを取得するタスクが実施された。プロジェクト開始当初、一般歯科医が最大限の恩恵を得るために、製品は歯科咬翼法 X 線の専門家の分析を模倣しなければならないと判断されていた。（顎顔面放射線科医は、歯科医療画像を分析する臨床の専門家であり、咬翼法 X 線でエナメル質のみの隣接面の齲歯を早期に発見し、他の病態や画像アーチファクトと区別することに長けている。）このため、5 名の口腔顎顔面放射線科医を採用し、各医師がすべての画像を分析し、隣接面の齲歯の位置を国際的に認められた 4 段階のグレーディングスケールを用いて記録した。専門家の分析を統合することで、一つの「ゴールドスタンダード」データセットが提供された。

この事例研究では、歯科咬翼法 X 線のチェアサイドでの分析のためのアプリケーション開発について説明する。

X 線画像で小さな病変を見つけることは困難であるため、ML サブシステムは画像の大きな特徴を利用するアルゴリズムのパイプラインとして設計された。このプロトタイプは、シンプルで双方向性の製品デモのバックボーンを形成し、ビジネスのコラボレーターや潜在的な顧客が入力として独自の画像を提供し、結果を評価できるようにした。これは、その後の開発の指針となる貴重なユーザーフィードバックとなった。また、ソリューションが、すべての収集ハードウェアからの画像に対して汎用的である必要性が示され、そのためプロ

ジェクトは（プロジェクト/製造のフェーズで発生した）一般診療データを収集する必要があった。

プロトタイプ性能報告書には、故障モードの図説を含む、定量的な性能測定と定性的な評価で実行された全ての実験とともに、トレーニングや評価方法の詳細な説明が含まれていた。また、この報告書には製品の性能に関するよく検討された予測も含まれていた。（知的財産分析により、保護を検討すべき新規 IP の運用ならびに識別をする上での自由度を決定した。）

プロトタイプの結果、他のサイトから収集した X 線では、アルゴリズムの性能が低い事が示された。その後、倫理的に承認された臨床データ収集プロジェクトが開始され、10 の一般歯科診療所から画像を収集し、顎顔面放射線科医のパネルによって注釈が付けられた。収集され、注釈が付けられると、画像の 20%から成るテストセットが無作為に選ばれ、モデルの全てのトレーニングとバリデーション、およびパイプラインの評価から除外されたサイトで層別化され、最終的な ML サブシステムと最終製品の評価ための偏りのないデータセットが提供された。

プロトタイプ実験に基づき、本事例研究における ML サブシステムの設計は 3 つの異なるコンポーネントから構成されている。(a) パターン認識の歯検出器、(b) 動的計画法による齲歯の候補位置スーパーセットの特定、(c) 候補を早期齲歯またはその他のいずれかに分類する深層学習モデル。これらの構成要素のリスク分析により、最終段階が最もリスクが高い事が判明した。これは、パターンマッチングと比較してニューラルネットワークを使用するという革新的なアプローチであることと、十分な検出性能を確保するために大量のトレーニングデータセットが必要となる可能性があるためだ。このため、データ消費量の少ない代替分類器を用いたり、追加データを取得したりなどの緩和戦略が開発された。

歯検出器と齲歯分類器の両方に ML モデルが含まれていたが、簡潔にするため、最後のコンポーネントである深層学習分類器の計画について説明する。Python 言語と開発環境は、ML モデルの迅速なプロトタイピングをサポートし、サードパーティーの畳み込みニューラルネットワークサポートライブラリを利用できるため、深層学習分類器の開発に選ばれた。

データ形式はコンポーネント毎に指定されている。最後のコンポーネントに注目すると、入力は齲歯を示すものとして前のコンポーネントで生成された隣接面の歯縁に沿った候補位置のセットであり、出力は候補が齲歯を有すると分類されるかどうかの測定確率として指定される。これらの候補位置と信頼度は、齲歯注釈の専門家のゴールドスタンダードと比較することで評価できる。モデルの出力は、距離の許容範囲内で専門家の齲歯注釈と同じ隣接面縁に存在する場合、真陽性に分類された。これにより、モデル性能を評価および比較するための主要な性能指標として使用される曲線下面積（area under the curve; AUC）を用いて、全ての候補位置および全ての潜在的な閾値について蓄積された受信者動作特性（receiver operator characteristic; ROC）曲線の構築が可能になった。

歯の隣接面縁にある早期齲歯の物体検出という課題に対して、2 つの候補となる機会学習

アルゴリズムが特定された。(a) ランダムフォレスト分類と(b) 深層学習分類器である。

どちらも表面縁に沿った候補位置のセットとして入力データが必要であった。グランドトゥルースデータは、各ポイントが非齲歯または齲歯であるかどうかの分類と、それに続けて齲歯領域をエネメル質齲歯（すなわち、歯の外側のエナメル質のみに浸透した齲歯）あるいいは象牙質齲歯（歯の象牙質へさらに進行した齲歯）へのサブ分類も合わせてできている。性能指標はエナメル質齲歯の分類のための ROC 曲線下の領域として決定された。

モデルの開発は、反復的なトレーニング、テスト、評価、ハイパーパラメータのチューニングの通常の ML ライフサイクルに従った。画像ソースサイトを元に層別化した 5 分割交差検証が使用され、データをトレーニング用と検証用のデータセットへ分割した。各反復のハイパーパラメータと結果の注意深い記録を含むハイパーパラメータのチューニングのアプローチは、性能にプラスの影響を与えた変更を手動で特定し、それに応じて次の反復のためにパラメータを微調整し、性能指標を比較し、結果の定性的評価を実行することであった。

実験が進むにつれ、深層学習分類器ソリューションがランダムフォレスト分類よりも優れた性能を提供することが明らかになり、そのモデルに最適なハイパーパラメータを選択することに焦点が向けられた。

製品仕様では、インターネットの接続や特注のハードウェアや追加ソフトウェアを使用せずに、通常の PC 上で製品を動作させる必要があった。この非機能的な要件を満たすために、Python で開発された ML サブコンポーネントの推論モジュールは、.net API で拡張されたパイプラインの C++コンポーネントに移植された。評価、モデルのトレーニング、ハイパーパラメータのチューニングに関する全てのモジュールは除外している。深層学習モデルは Python（TensorFlow）から ONNX 形式に移植され、ランタイムの推論コードは Microsoft ML サポートライブラリを使用して C#で書かれた。製品の全ての設計、メンテナンス、リリース活動は、ISO13485 規格に照らして監査された。

本製品では、推論モジュールに加え、ランタイム推論の性能を検証するコードがビルドおよびテストパイプライン環境に移植された。これにより、コードビルドサイクル中のモデル性能の自動テストとバリデーションが可能になった。テストとリリースのプロセスに完全に統合された性能評価レポートを開発するという設計上の決定が下された。インターフェースは、テスト画像を提供し、その結果をゴールドスタンダードの注釈と比較する機能によって増強された。リリースパイプラインのテストコンポーネントに統合され自動化されたテストを推進するために、Selenium UI テストが使用された。モデルのトレーニングとハイパーパラメータのチューニング機能は、製品の中核とはみなされず、ソフトウェアのメンテナンスと性能のバリデーションプロセスから除外された。

製品の性能の追加バリデーションを提供するため、製品を使用することでエナメル質のみの隣接面齲歯を検出する歯科医の能力が強化されたかどうかを調査する臨床試験が倫理的に承認された。この研究では、本製品を使用した歯科医は 75.8%の早期齲歯を発見したのに対し、AI の支援なしでの咬翼 X 線画像を使用した歯科医はわずか 44.3%の検出率であり、

感度 71%で統計的有意な増加が報告された[3]。

研究やその後のアーリーアダプターでの使用中に、製品から最大限の恩恵を得るためにはユーザーをトレーニングおよび教育する必要があることが明らかになった。この種の対話型 AI システムでは、臨床医が AI ワークフローループの不可欠な部分であり、人間は AI システムが提案する関心領域を調べる必要があるが、それらを盲目的に真実として受け入れることはない。代わりに、ユーザーは診断の決定を下す時には臨床的知識と判断を生かした。歯科医が患者の診断と治療をより適正に行えるよう、本製品の特異度と感度の性能指標ならびに出力を最良の方法で解釈するための詳解をユーザーに提示するため、トレーニング資料が作成された。

結論

この記事では、SaMD 製品を例に、規制対象のライフサイエンスにおける ML コンポーネントやサブシステムのライフサイクル活動について検討した。また、より広いシステムやアプリケーションのライフサイクルの中で ML サブシステムやコンポーネントのライフサイクルや管理について説明した。このような ML の使用は、創薬や臨床開発から、認可後の製品監視やリアルワールドデータ分析に至るまで、医薬品のライフサイクル全体を通じて行われている。

本文以上

図 1：ML サブシステムライフサイクル

図 2：製品のグラフィカルユーザーインターフェース[3]

（P16 左の枠内）

定義

人工知能 (Artificial intelligence; AI): 具体的な目的を達成するために、（ある程度の自律性を持って）それに関する状況を分析し行動をとることによって知的行動を示すシステム。AI ベースのシステムは、仮想空間で動作する純粋なソフトウェアベースにすることも、ハードウェアデバイスに組み込むこともできる。科学分野としては、AI は機械学習（深層学習と強化学習がその具体例である）、機械推論 (計画、スケジューリング、知識の表現と推論、検索、最適化など) 、ロボティクス（サイバーフィジカルシステムへの他の全ての技術の統合ばかりでなく、制御、知覚、センサー、アクチュエーターをも含む）などのいくつかのアプローチと手法が含まれる。

機械学習 (Machine learning; ML): AI の一分野であり、（トレーニングデータのような）入

力データから予測モデルを立てる（トレーニングする）プログラムあるいはシステム。そのシステムは、学習したモデルを使用して、モデルのトレーニングに使用されたものと同じ分布から抽出された新しいデータから有用な予測を行う。

深層学習: 深層構造学習または畳み込みニューラルネットワーク(convolutional neural networks; CNNs)とも呼ばれ、表現学習による人工ニューラルネットワークに基づく機械学習法のファミリーの一部である。

ランダムフォレスト: 回帰および分類の問題を解決するために用いられる ML 手法。

ケースデータ: ML によって処理される情報のタイプを代表し、偏りなく戦略的に選択されたデータであり、トレーニングとバリデーションのサンプル／サブセットの選択に使用される。

トレーニングデータ: モデル／分類子のモデルパラメータを適合させるために、学習に用いられるサンプルおよび／またはサブセット。

バリデーションデータ: モデルを評価するためモデルのトレーニングとチューニングの際に使用されるデータのサンプルあるいはサブセット。データはトレーニングデータからは独立した評価を提供するが、モデルのトレーニングプロセスから完全に独立してはいない。データサイエンスと AI/ML では、バリデーションは GxP コンピューター化システムにおいて異なる使われ方をしている。

テストデータ: 全てのトレーニング、チューニング、およびバリデーション活動から除外されたデータのサンプルおよび/またはサブセットであり、完全に特定されたモデル/分類子の性能を算定し評価するために用意されている。

ゴールドスタンダード/「グラウンドトゥルース」: モデル/分類子の出力が最終的に評価および/または比較され、承認された外部基準としての役目を果たす一連の結果。