クラウド上の統合環境を利用したデータ分析と最適化
—Watson Studio—

赤石　雅典，岸代　憲一，米沢　隆

IBM 社は，クラウド上に統合データ分析環境として Watson Studio を提供している．今回，この Watson Studio に新しいサービスとして従来 CPLEX として知られていた製品が，Decision Optimization という名称で利用可能になった．本稿では，Watson Studio と CPLEX の概要を紹介した後，Watson Studio 上で Decision Optimization を利用する方法について解説する．最後にこのサービスを活用した想定事例の紹介をする．

キーワード：最適化，数理計画法．機械学習，クラウド，統合分析環境

1.　データ分析と最適化を使った業務システム開発の課題

データ分析と最適化を使った業務システム開発のタスクは，大きく次の三つと考えられる．

1. 検証 (PoC(Proof of Concept))
   想定したユースケースで意味のある精度が出るかを確認．
2. 開発
   PoC の次のステップとして実業務データを学習データとして本番目的の機械学習モデルを開発．
3. 本番
   本番用に開発した機械学習モデルを呼び出して使うアプリケーションを構築．

それぞれの局面での要件をまとめると表 1 のようになる．

従来，これらのタスクはまったく別個のものであり，それぞれ別の開発環境・ツールを利用するのが通常であった．このため，end-to-end で見た開発工程は，相当煩雑で工数のかかるものとなってしまっていた．

この課題に対応するための統合開発ツールが Watson Studio である．Watson Studio は，共通の UI によって，三つそれぞれのタスクのすべての領域をカバーしつつ，個々の領域において効率のいいアジャイル型のモデル開発環境を実現している．2 節でその概要を説明する．

あかいし　まさのり，きしろ　けんいち，よねざわ　たかし
日本アイ・ビー・エム株式会社
〒 103–8510 東京都中央区日本橋箱崎町 19–21
akaishi@jp.ibm.com
kishiro@jp.ibm.com
yonezat@jp.ibm.com

2.　IBM Cloud と Watson Studio

IBM Cloud は，IaaS，PaaS，SaaS などさまざまなレベルのサービスを統合的に提供しているクラウドサービスである．Watson Studio は，IBM Cloud の 1 サービスで，データ分析や機械学習モデル構築のための統合プラットフォームとして提供されている．

図 1 に IBM Cloud 内の Watson Studio 全体像を示す．

2.1　Watson Studio と関連のあるサービス

本節では，Watson Studio と関連のあるクラウドサービスを紹介する．

2.1.1　Watson API

Watson API は，最初に Watson サービスが商用化された時から提供されている API サービス群である．主に，テキスト分析，画像，音声など，非構造化データを対象とした事前構築済みモデルとなっている．

2.1.2　Watson Konwledge Catalog

Watson Konwledge Catalog では，データを機械学習モデルの入力とするためのツール群が提供されている．パスワード・URL などのデータベース接続情報（他クラウド DB を含む）や，その配下のテーブルをカタログする機能や，データ整形をバッチ処理で行う Refinary などの機能がある．図 2 は，テーブルをカタログ登録中の画面，図 3 は，登録後の様子である．図 4 には，登録後のリンクをクリックしてデータ内容をプレビューしている画面を示した．

いったん DB のテーブルをカタログ登録しておくと，このテーブルは SPSS Modeler（2.2.2 節で説明），あ

表 1　検証・開発・本番環境の整理

	検証	開発	本番
モデル	多数のモデルの試行錯誤	検証結果に基づくモデル実装	実運用に基づく改善
データ	検証用データ・本番相当データ	本番相当データ	本番データ
プロセス	単機能での実装	サービスモデルの実装・結合テスト	複数アプリの動的連携
テスト	精度・処理時間の確認	システム開発としての信頼性確認	本番での実績の蓄積と検証
管理	アジャイル	世代管理	精度の監視・モデルの管理

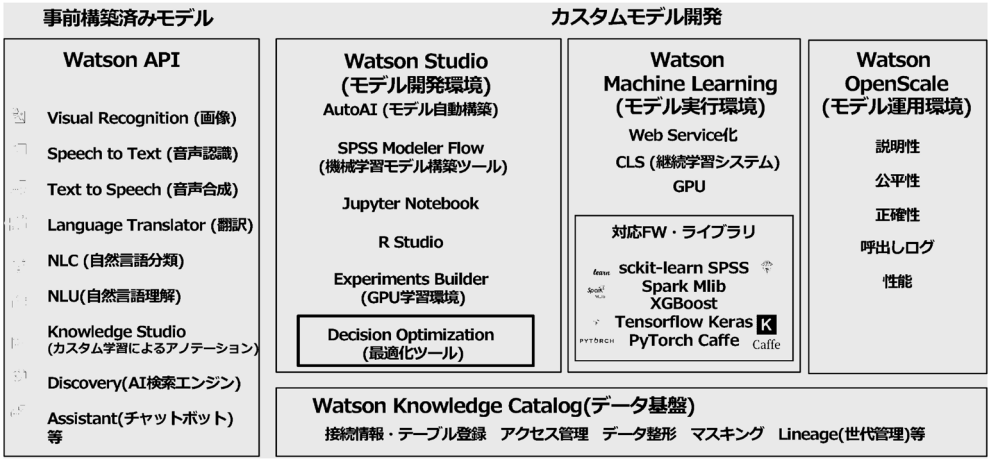

図 1　IBM Cloud 全体図（Watson Studio 関連）

図 2　テーブルをカタログ登録中の画面

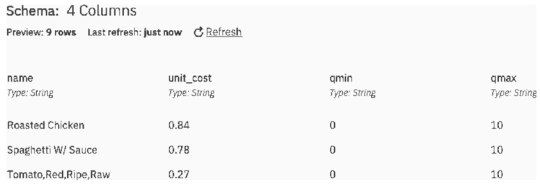

図 4　テーブルデータプレビュー結果

図 3　カタログ登録後の様子

るいは Model Builder（4.2 節で説明）のデータソースとして直接参照することが可能である．1 節で紹介した開発プロセスとの関連で言うと 2 番目の本番用モデル開発のため，DB 上にある業務データを学習データとして直接利用するためのツールという位置付けになる．

2.1.3　Watson Machine Learning

Watson Studio がカスタムの機械学習モデルの開発環境であるのに対して，その実行環境は Watson Machine Learning となる．二つのサービスは UI 画面を通じてシームレスに連携されているため，開発者はサービスの違いを意識することなくモデル開発・テスト実行などを行うことができる．1 節の開発プロセスとの関連で言うと 3 番目の本番用アプリケーション開発で役立つ機能である．

2.1.4　Watson OpenScale

Watson OpenScale は本番稼働中の機械学習モデルの運用を支援するサービスである．具体的な機能としては，「説明性」「公平性」「正確性」の確認・監視や，性能管理機能などを持っている．利用フェーズでいうと，本番用アプリケーションの運用フェーズで活用できる機能となる．

2.1.5　Spark

IBM Analytics Engine というサービスは，Spark / Hadoop の実行環境を提供する．いわゆるビッグデー

タ分析に利用できるサービスとなる．エンジンの起動は Watson Studio の Jupyter Notebook から可能となっている．

2.2　Watson Studio の主要サービス

Watson Studio は，機械学習モデルの開発環境だが，その内部はさらに多くのサービス群から構成される．その代表的なものを紹介する．

2.2.1　AutoAI

AutoAI は，学習対象の CSV データをクラウド上にアップロードし，目的変数列を指定するだけで，後は全自動で高精度の機械学習モデルを生成するツールである．現在は分類型と回帰型の教師あり学習モデルに対応している．自動化のプロセスには，データ前処理，最適モデルの選定，ハイパーパラメータチューニング，特徴量抽出が含まれている．

2.2.2　SPSS Modeler Flow

30 年の歴史をもつデータ分析・予測ツールである SPSS Modeler の簡易版が，Watson Studio の 1 機能としてクラウド上で動くようになっている．予測モデルを作るのでなく，データ前処理をプログラムなしで行えるツールとしての使い方も可能である．また，クラウド版固有の特徴として，予測モデルの Web サービス化が簡単にできる点がある．

2.2.3　R Studio/Jupyter Notebook

データ分析をプログラミングで行う上級データサイエンティスト向けに，R Studio と Jupyter Notebook の環境も Watson Studio 上に提供されている．Jupyter Notebook では Python だけでなく R 言語も選択可能である．Jupyter Notebook 作成時には，事前導入済みのライブラリの選択が可能で，ライブラリを適切に選ぶことで，Spark API や，Decision Optimization API（4.1 節で紹介）をすぐに利用することができる．

3.　CPLEX / Decision Optimization

本節では，最適化の領域で多くの実績をもつ CPLEX の紹介を行う．

3.1　CPLEX の歴史

CPLEX は元々は Robert E. Bixby によって開発され，1988 年に Optimization 社により CPLEX として商業的に販売された．同社は 1997 年に ILOG 社に買収され，ILOG 社は 2009 年 1 月に IBM 社に買収されたため現在は IBM 社の製品となっている(表 2) [1].

表 2　CPLEX の歴史

1988 以前	Robert Bixby が C 言語で実装
1988	CPLEX 社を創業 (CPLEX 1.0)
1992	CPLEX 2.0
1997	ILOG 社が CPLEX 社を買収
2009	IBM 社が ILOG 社を買収

3.2　2 種類のモデル

CPLEX は数理計画法 (Mathematical Programming, MP) と呼ばれるモデルと制約プログラミング

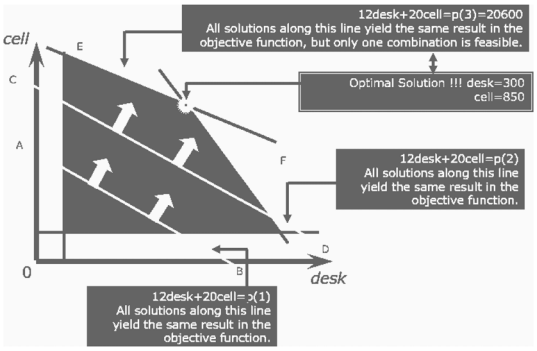

図 5　数理計画法解法例

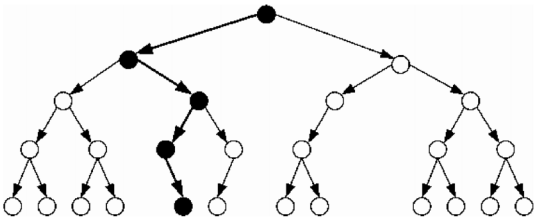

図 6　制約プログラミング解法例

(Constraint Programming, CP) と呼ばれるモデルの 2 種類のモデルをもっていて，用途によって使い分ける形になる．それぞれのモデルの特徴を簡単に説明すると，以下のようになる．

3.2.1　数理計画法

線形計画法、整数計画法等にマッピングできる問題に適する(図 5)[2].

- 数百万個の制約および変数をもつ問題を解くことのできる強力な計算エンジン
- 連続的な変数の関係から，最大・最小となる組み合わせの境界を高速に計算する
- 適用領域：資源割り当ての最適化など

3.2.2　制約プログラミング

制約が複雑な問題やスケジューリング問題に適する(図 6) [3].

- 経路決定や詳細なスケジューリングといった線形ではなく離散的な問題に対応
- 適用領域：スケジューリング，要員配置，ダイヤ

グラム作成などの最適化

3.3 開発環境・言語

開発環境は CPLEX Optimization Studio (COS) と呼ばれる Eclipse のプラグインが使われる．開発言語としては OPL (Optimization Programming Language) と呼ばれる CPLEX 専用の言語を利用することが多いが，最近は Python API も利用されつつある．以下に OPL のコードサンプルを記載する．

```
# 決定変数：正整数 x, y
dvar int+ x;
dvar int+ y;

# 目的関数（最大化）：2x+3y
maximize x*2+y*3;

# 制約：4x+3y<=120, x+2y<=60
subject to{
 x*4+y*3<=120;
 x+y*2<=60;
};
```

4. Watson Studio における Decision Optimization

本節では Watson Studio で新たに可能になった Decision Optimization の呼び出し方を説明する [4]．呼び出し方は

1. Jupyter Notebook 上の Python API を利用する
2. Model Builder の利用
3. Watson Machine Learning 上の Web サービスとしての利用

の三つがある．Python で実装した場合そのすべての方法が，OPL で実装した場合は 2 と 3 の方法が利用可能である．

4.1 Jupyter Notebook 上の Python API

Watson Studio で新規 Jupyter Notebook 作成時に「Default Python 3.6 XS + DO」というランタイムを選択すると自動的に機能制限のない Python API が利用可能な環境となる．（上記ランタイムを利用しない場合，ルール数最大 1,000 個という機能制限がかかる．）ちなみに DO は Decision Optimization の略である．以下に 3.3 節で紹介した OPL 版と同じ機能実装を Python API 版として実装したコーディングサンプルを示す．

```
# ライブラリのimport
from docplex.mp.model import Model

# モデルオブジェクトの生成
mdl = Model()

# 決定変数：正整数 x, y
x = mdl.integer_var(lb=0, name='x')
y = mdl.integer_var(lb=0, name='y')
```

Products 3 rows

	name String	demand Number	insideCost Number	outsideCost Number
1	kluski	100	0.6	0.8
2	capellini	200	0.8	0.9
3	fettucine	300	0.3	0.4

図 7 Model Builder 入力データ例

```

# 制約
mdl.add_constraint(4*x+3*y<=120)
mdl.add_constraint(x+2*y<=60)

# 目的関数
mdl.maximize(2*x+3*y)

# 最適化の実施
mdl.solve()
```

4.2 Model Builder の利用

Watson Studio の *Add to project* のメニューから *Decision Optimization model* を選択すると Model Builder が呼び出される．実装コードは OPL と Python が選択可能である．

Model Builder 利用時には，入力と出力データの形式に注意する必要がある．どちらに関しても，表形式にする必要がある．表の実体は CSV ファイルであっても，Data Catalog で登録されたテーブルであっても構わない（図 7）．

以下に，先に説明した OPL によるモデル開発の場合に，図 7 の表形式データを読み込むためのコード実装例を示す．実装上の注意点として，入力ファイルの名称を “Products” と，OPL コード上の変数名と合わせることが必要である．

```
tuple TProduct {
key string name;
float demand;
float insideCost;
float outsideCost;
};
{TProduct} Products =...;
```

4.3 Watson Machine Learning 上の Web サービスとしての利用

Decision Optimization を Watson Machine Learning 上の Web サービスとして利用するためには，4.2 節で動作するようになった Python または OPL のコードを，gz 形式で圧縮した後，下記のような Watson Machine Learning API を利用したコードで Watson Machine Learning にモデルとして登録する．

```
model_metadata = {
 client.repository.ModelMetaNames.\
NAME: "Diet",
 client.repository.ModelMetaNames.\
DESCRIPTION: "Model for Diet",
 client.repository.ModelMetaNames.\
TYPE: "do-docplex_12.9",
 client.repository.ModelMetaNames.\
RUNTIME_UID: "do_12.9"
}

model_details = client.repository.\
store_model(model=\
'/home/dsxuser/work/model.tar.gz',\
 meta_props=model_metadata)

model_uid = client.repository.\
get_model_uid(model_details)

print( model_uid )
```

登録に成功すると，deployments.create 関数呼び出しで Web サービス化することができる．Web サービス化されたモデルは，ジョブ投入することでモデル呼出しが可能になる．以下にジョブ投入時のサンプルコーディング例を示す．

```
 client.deployments.\
DecisionOptimizationMetaNames.\
INPUT_DATA: [
 {
 "id":"diet_food.csv",
 "values" : diet_food
 },
 {
 "id":"diet_food_nutrients.csv",
 "values" : diet_food_nutrients
 },
 {
 "id":"diet_nutrients.csv",
 "values" : diet_nutrients
 }
 ],
 client.deployments.\
DecisionOptimizationMetaNames.\
OUTPUT_DATA: [
 {
 "id":".*\.csv"
 }
 ]
}

job_details = client.deployments.\
create_job(deployment_uid ,\
solve_payload)
```

4.4　3 方法の使い分け

最適化問題を含んだ業務アプリケーションに関してもその開発工程は 1 節で紹介した機械学習モデルによる業務アプリケーション開発工程とほぼ同じと考えられる．

4.1 節の方法は，簡単に試せてデバッグ・コード修正が容易という観点で一つめの検証工程 (PoC) に最適である．

4.2 節の方法では，データソースに業務テーブルを指定できることから，二つめの開発工程に向いている．

4.3 節の方法の特徴は，耐障害性やスケーラビリティとなる．Watson Machine Learning の実装は，Kubernates ベースのサービスになっているため，同時利用時の高負荷や障害時の対応を基盤側が自動的に提供し，ユーザーが特に意識する必要がない．コード開発の手間は一番かかるが，こうした点は開発工程 3 番目の，本番工程において大きなメリットとなる．

ユーザーが実際に利用する際には，これらの特性を理解した上で三つの開発方法を使い分ける形になる．

5.　Watson Studio 上の Decision Optimization 活用案

4 節までで説明したように，Watson Studio は SPSS Modeler や Jupyter Notebook そして Decision Optimization など複数のサービスを共通基盤上にもっており，複数のサービスを複合した高度なソリューションを Watson Studio 上で実装可能である．以下では，実現可能性のある複合ソリューション案を例示する．

5.1　個人の購買予測 + マーケティング戦略最適化

本節で紹介するのは，実際の事例があり，また Decision Optimization のサンプルアプリも公開されているケースである．具体的な内容はサンプルアプリ [5] に基づいて説明する（図 8）．

5.2　問題の定義

銀行業務のマーケティングのユースケースを想定する．営業対象はすでに銀行の口座をもっている顧客で，顧客属性と過去の営業実績履歴は業務データとしてもっている（図 9）．銀行は複数の商品（住宅ローン，定期預金，年金）と販売チャネル（セミナー，プレゼント，メール）をもっていて，1 人の顧客には最大一つの商品，一つのチャネルで顧客に対する営業を行うこととする．ここで解くべき問題は「限られた予算の範囲内で商品購買の期待値を最大化するにはどのような個別アプローチを行ったらいいか」である．

5.3　予測モデルの構築

顧客属性を入力データに，営業実績履歴を正解データに予測モデルを構築する．正解データは購買した・しないの二値なので二値分類モデルになるが，モデルの方式を選定することで，購買確率を出力とすることも可能である．以下の議論はこの前提で進めることとする．

5.4　全顧客に対する購買行動予測

モデルができると，顧客マスターを入力データとすることで全顧客に対する購買行動予測が可能となり，

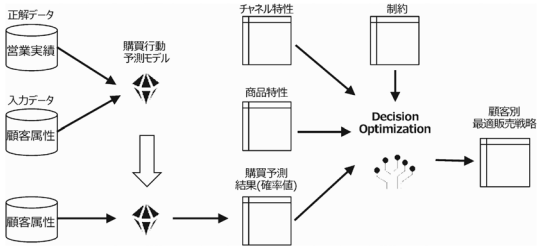

図 8　銀行マーケティング事例

	customer_id	age	age_youngest_child	debt_equity	gender	bad_payment	gold_card	pension_plan
0	15	45	12	45	0	0	0	0
1	16	43	12	43	0	0	0	0
2	30	23	0	23	0	0	0	0
3	42	35	8	35	1	0	0	0
4	52	43	12	43	1	0	0	0

図 9　顧客属性

	id	Savings	Mortgage	Pension
0	44256	0	0	0
1	46883	0	0	0
2	32387	0	0	0
3	25504	0	0	0
4	35979	0	1	0
5	29822	0	0	0
6	36731	0	0	0
7	8756	0	0	0
8	112583	1	0	0
9	18143	0	1	0

図 10　顧客別販売戦略

結果は確率値で得られる．

5.5　最適化モデルの構築

販売チャネルの特性と商品の特性はあらかじめわかっているとする．この場合，これらの特性と前のステップで得られた顧客購買予測結果を入力として，目的関数を「商品販売期待値の最大化」とする最適化問題を構成することが可能である．この際，営業費用の総予算額や「同一顧客には最大 1 チャネル 1 商品の営業しか行わない」というルールが制約ということになる．

5.6　最適化モデルの出力

前節までで説明した最適化問題を解いた結果は，顧客別の販売チャネル，販売商品の一覧となる．個別顧客に対応して最適化された販売戦略が得られたことになる（図 10）．

6.　まとめ

本稿では 1 節で，機械学習モデルを利用した業務アプリケーション開発の課題を提示し，2 節で，その解決策としてのクラウドサービスである Watson Studio の紹介を行った．

3 節では，30 年の歴史をもつ最適化エンジンである CPLEX の紹介を行い，4 節で Decision Optimization として Watson Studio に統合されつつある CPLEX の実装方法を説明した．

5 節ではこの新しい枠組みで可能となる複合ソリューションの例を提示した．

Watson Studio は，このような機械学習モデルと最適化ソリューションを組み合わせた新しい業務アプリケーションを開発するための統合プラットフォームとして最適なものである．

CPLEX は，他の分析ソリューションとの統合により，今後ますます活用例が広がると考えられる．

参考文献

[1] Carnegie Mellon University, IBM ILOG CPLEX What is inside of the box?, http://egon.cheme.cmu.edu/ewo/docs/rlima_cplex_ewo_dec2010.pdf （2019 年 12 月 17 日閲覧）
[2] IBM, Tutorial: Linear Programming, https://github.com/IBMDecisionOptimization/tutorials/blob/master/jupyter/Linear_Programming.ipynb （2019 年 12 月 17 日閲覧）
[3] IBM, Planning/Scheduling with CP Optimizer, https://cp2019.a4cp.org/PDFs/P-Laborie.pdf （2019 年 12 月 17 日閲覧）
[4] IBM, Decision Optimization, https://dataplatform.cloud.ibm.com/docs/content/DO/DOWS-Cloud_home.html （2019 年 12 月 17 日閲覧）
[5] IBM, Promoting financial products to bank customers, https://github.com/IBMDecisionOptimization/DOforDSX-MarketingCampaigns-example/blob/master/jupyter/MarketingCampaigns.ipynb （2019 年 12 月 17 日閲覧）