# データ分析プロセスの概要

分析プロセスの代表例は下表の通りである．必ずしも順に進んでいくものではないが，

|  |  |
| --- | --- |
|  |  |
| 課題設定 | 解決したい問題を明確にし，データ分析の課題に落とし込む |
| データ収集 | 課題に必要なデータを集める |
| 探索的データ分析 | データの特徴や概要を掴み，分析の方針や仮説を立てる |
| モデル構築・評価 | モデルを作り，性能を評価しながら強化する |
| モデル運用 | 構築したモデルを実務に組み込んだり，分析から得られた知見を顧客に報告する |

# 課題設定

データ分析の最初のプロセスは課題設定である．分析自体は目的ではないため，解決したい実務上の課題を明確にして，データ分析の課題に落とし込むことは重要なプロセスである．課題設定時のポイントは主に以下2点である．

## 分析の目的

データ分析の目的として，大別すると2つの考え方がある．課題がどちらに該当するのか，あるいはその両方なのか，どちらをどの程度優先するか等を意識した上で，分析手法を検討することが重要である．

* 理解志向

「データがどういうメカニズムで生成されているのかを理解する」ことを第一に目指すモデリング

* 対象となる現象において，どの要因が強く影響を与えているのかを特定する
* なぜそのような現象が起こるのかを明らかにする
* 応用志向

手元にあるデータをもとに，未知のデータに対して予測

## 分析タスクの設計

回帰問題か分類問題に帰着できるか

* + 回帰：連続値（数値）を予測すること
  + 分類：離散値（クラス）を予測すること
* 必要なデータセットを用意できるか

# データ収集

予測する数値やカテゴリを目的変数と呼ぶ．また，目的変数を予測するために使う値を特徴量と呼ぶ．なお，特徴量は，変数や説明変数と呼ぶこともある．今回はこれらの名称を特に区別せずに用いる．

# 探索的データ分析

課題が設定できたら，分析の第一歩として，探索的データ分析（Explanatory Data Analysis，EDA）を行う．この目的は諸説あるが，主なものとしては以下の通りである．

* データの傾向と特徴を理解する
* データ間の関係性を理解する
* 予測に有効な特徴量を見つける
* データから不要な情報を取り除く

探索的データ分析は大別すると以下5つのプロセスで構成させるが，常に順番通りに進むわけではなく，各プロセスを行き来しながら，最終的な目標地点であるデータの選択を行うことになる．

《図》

[1]データの理解 → [2]クリーニング → [3]分析の切り口の検討 → [4]データの分析 → [5]データの選択

## データの理解

### データ確認と仮説検討のポイント

データセットの背景やその分野の知識があるとは限らない．各データの意味が分からなければ，何が重要なデータが分からない．そこで，まずは以下の事項について確認する．

* 確認事項
* データセットの概要
* データの意味
* データの種別
* 要約統計量（値の範囲等を含む）

データを確認した上で，例えば以下のような観点で仮説を立てることで，後続プロセスで少しずつ見えてくる．

* 仮説検討の観点
* 目的変数と関係がありそうなデータ・関係なさそうなデータは何か？
* 優先的に分析をしたほうがよさそうなデータはあるか？
* まずは意味から調べる必要があるデータはあるか？
* どのデータ間の関連性が強そうか？

### データ種別

分析の基礎としてまず知っておくべき種別として，①質的・量的データ，②尺度分類を紹介する．

* 量的データ・質的データ
* 質的データ

カテゴリデータともいう．性別の「男性 / 女性」，評価の「良い / 普通 / 悪い」のようにデータの値に意味があり，分類や種類を区別するデータのこと．数値で測定することができないため，四則演算を行うことができない．

* 量的データ

数値によって計測・集計・分析ができ，四則演算が可能なデータのこと．その中でも，身長や気温のように連続値をとる連続データ，あるいは人数や個数のように離散値をとる離散データがある．

注意点：

データが数値であってもカテゴリデータとなる場合がある．例えば，「0:A型 / 1:B型 / 2:O型」「0:良い / 1:普通 / 2:悪い」といった評価のように値に質的な意味がある場合はカテゴリデータになる．

* 尺度分類

データの性質により4つに分類される．低い水準から①名義尺度，②順序尺度，③間隔尺度，④比例尺度であり，高い水準の尺度は低い水準の尺度の意味を含むようになっている．比例尺度は，下位の3つの意味も持ち合わせており，下位の尺度へ変換が可能である．逆に下位の尺度から上位の尺度に変換することはできない．

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 尺度分類 | 概要 | 可能な計算 | 主な統計量 |
| 質的データ （カテゴリデータ） | 名義尺度 | 他と区別するためのもの  （例：性別，血液型） | なし | 最頻値 |
| 順序尺度 | 順序には意味があるが，間隔には意味がない  （例：大学のGPA） | 大小比較のみ | 四分位数 |
| 量的データ （数値データ） | 間隔尺度 | 順序，差に意味はあるが，比率に意味はない．0は相対的なもの  （例：入試の偏差値） | 加算，減算 | 平均値  中央値  標準偏差 |
| 比例尺度 | 順序，差，比率に意味があり，0が絶対的な意味をもつ  （例：体重） | 四則演算 | 平均値  中央値  標準偏差  変動係数 |

### 要約統計量

要約統計量とは，データの特徴や傾向を代表的に表す値のことである．大量のデータをそのまま眺めても，特徴が掴みにくいため，平均値，中央値，標準偏差といった統計量を見ることでデータの特徴を掴むことが出来る．

pandasでは，describe関数で主要な統計量を出力することができる．ただし，データ種別（量的データ・質的データ）に応じて表現できるものが異なることに注意されたい．以下がpandasのdescribe関数で出力される統計量である．

|  |  |  |
| --- | --- | --- |
| 統計量 | 対象のデータ種別 | 概要 |
| count | 全データ | データの個数．欠損値はカウントされない． |
| unique | 質的データ | 一意な要素の個数 |
| top | 最頻値 |
| freq | 最頻値の頻度 |
| mean | 量的データ | 平均値（算術平均） |
| std | 標準偏差 |
| min，max | 最小値，最大値 |
| パーセンタイル点 | デフォルトでは25%，50%，75%の位置を出力．  引数percentileで変更可能．50%が中央値である． |
| first | 日付データ | 最初の日 |
| last | 最後の日 |

## クリーニング

ほとんどの場合，データセットには欠損や分析に適さないデータが含まれている．そのまま分析してしまうと，解釈を誤るほか，モデル学習に影響を及ぼすことがある．ここでは，代表的な処理として，以下4つを紹介する．

|  |  |
| --- | --- |
| 方法 | 概要 |
| 欠損値の処理 | 欠損値を見つけて，削除等を行う |
| 外れ値の処理 | 外れ値を判定し，削除・クリッピング等を行う |
| 表記揺れの統一 | “男性”，”男”のような表記揺れを統一する |
| 重複データの削除 | 重複行，重複列を削除する |

### 欠損値の処理

データに欠損値がある場合，データの特徴や傾向を誤って解釈してしまう可能性がある．例えば，Pythonのライブラリ（pandasやseaborn等）を使って，統計量の確認や可視化を行う場合，欠損値は除外されて計算や可視化が行われることが多い．欠損値が数件であれば影響は少ないが，多い場合はデータ本来の特徴を表しているとは言えない．処理方法はさまざまであり，主な手法は以下の通り．

* 行ごと削除
* 列ごと削除
* 定数による補完
* 固定カテゴリによる補完
* 最頻値による補完
* 予測値による補完

当日の課題中の分析では，欠損値処理済みのデータを使用するため，ここではこれ以上は解説しない．

### 外れ値の処理

データに外れ値がある場合，分析結果に大きく影響する場合がある．特に平均値や相関係数などは外れ値によって影響を受けやすいため，必ず確認する必要がある．

《四分位範囲を用いた外れ値処理の説明を加える》

外れ値の検出方法には様々なものがあり，データの特性や分析目的に応じて適切な方法は異なるが，ここでは，四分位範囲を用いた方法を解説する．

四分位範囲とは，四分位数は，データを小さい順に並べたときに，データを四分割するときの区切りの値を指す．小さい順に第一四分位

### 表記揺れの統一

表記揺れのデータが残っていると，データを正しく分類できない．表記揺れはいろいろなパターンがあるが，代表的なものとしては以下のようなものがある．

|  |  |
| --- | --- |
| 表記揺れのパターン | 例 |
| 全角・半角 | オオサカ/ ｵｵｻｶ |
| 漢字・カナ | 猫 / ネコ / ねこ |
| 送り仮名 | 取り組み / 取組 |
| 語尾 | ユーザー / ユーザ |
| 数字 | 5年 / ５年 / 五年 |
| 英単語 | スピード / speed |
| 固有名詞 | グーグル / Google |
| 大文字・小文字 | TOKYO / Tokyo / tyokyo |
| 省略形 | I can’t / I can not |

当日の課題では，表記揺れは処理済みのデータを使用する．

### 重複データの削除

## 分析の切り口の検討

データは見る視点や切り口によって，その特徴や解釈が大きく異なる．切り口の考え方には様々なものがあり，その全てを列挙するのは難しいが，ここでは以下3つを紹介する．これらの切り口を念頭におきつつ，後述の可視化や特徴量作成を行うと良いだろう．

|  |  |
| --- | --- |
| 方法 | 概要 |
| スライシング | 性別，年齢別といった層毎で表現する |
| 合成 | 複数のデータから新たなデータを作る |

### スライシング

データを様々な軸や水準で区切り，層化することで，意味付けされた特徴を理解することができる．スライシングの注意点として，層を細かく分けすぎると，本来は意味のない層でも意味のあるように見える分析結果となることがある．または，その逆で，本来意味のある層が意味のない層に見えてしまうこともある．

### ドリルダウン・ドリルアップ

### 合成

合成は複数のデータから新たなデータを作成することである．やり方としては，2つの質的データを掛け合わせて，新たな量的データ同士を四則演算で組み合わせることが多い．詳細は4.5で述べる．

## 可視化

データのクリーニングを行い，分析の切り口を決めたら，データの特徴や傾向，データ間の関係を理解するための可視化を行う．可視化を行う中で，疑問や仮説が生まれたら，再び前述のプロセスを行い，分析の視点や切り口を検討する．ここでは分析方法として以下6つを紹介する．

|  |  |
| --- | --- |
| 方法 | 概要 |
| ヒストグラム | データの広がり，分布から特徴や傾向を掴む |
| 棒グラフ | データの差や大小を把握する |
| 箱ひげ図 | データの位置や分布，外れ値を確認する |
| 散布図 | ２変数間の関係を詳細まで確認する |
| ヒートマップ | 多変数間の相関を確認する |

### ヒストグラム（１つのデータの分布）

ヒストグラムは，1つのデータの分布を把握するために使用する．どの範囲にどれだけの数のデータがあるのかを視覚的に表現することができる．ただし，区間と階級の決め方によって，見え方が異なってくることに注意されたい．また，欠損値があるとヒストグラムは作成できないため除外する必要がある．

### 棒グラフ（変数間の大小比較）

棒グラフは，複数のデータ間で値の大小を比較するために使用する．

### 箱ひげ図（変数間のデータ位置と範囲の比較）

箱ひげ図は，データの位置や範囲，外れ値を確認する方法である．最小値・最大値・中央値・第一 / 第二四分位数といった位置を確認することができる．また，四分位範囲による外れ値も視覚化できるため，外れ値の確認にも用いることができる．

### 散布図（2変数間の関係）

散布図は，2変数間のデータの分布や相関を確認するために使用する．後述のヒートマップでも相関を確認することができるが，散布図はデータ点ごとの位置まで細かく表現できることが特徴である．

### ヒートマップ

ヒートマップは，複数の変数間の相関を同時に視覚的に表現できる方法である．相関係数は量的データ同士でのみを算出できるため，ヒートマップで表現されるのは量的データの関係のみであり，質的データは除外されることに注意されたい．

《ヒートマップの例》

## 特徴量の作成

データセットの特徴や傾向，目的変数との関連が把握できたら結果をまとめる．モデル学習時において必要なデータ・不要なデータに分け，必要なデータは目的変数との関連度合いに応じて優先度を付ける．モデル構築時には，優先度の高いデータに対して，重点的に特徴量生成や分析を行う．

ビニング（）

順位への変換

### 量的データの変換

#### ビニング（ビン分割）

ビニング（ビン分割，binningとも呼ばれる）は，連続値を任意の境界値で区切り，カテゴリに変換する処理である．

例：年齢（22歳・55歳など）を年代（20代・50代など）に変換することで，年代ごとの傾向や特徴を捉えやすくなる．

演習：

#### 順位への変換

数値変数を大小関係に基づいた順位への変換する方法である．数値の大きさや間隔の情報をあえて捨て，大小関係のみを抽出する方法と言える．

例）乗船料を順位に変換する

### 質的データの変換

#### 規則性の探索

#### ダミー変数化

#### ラベルエンコーディング

### 変数の組み合わせ

#### 数値変数×数値変数

例：乗船料×年齢

#### カテゴリ変数×カテゴリ変数

例：大人かどうか×性別

# モデル構築

## ポイントと手順

* 次元の呪い

モデルの予測性能が変わらなければ，含まれる特徴量は少なければ少ないほど良い．特徴量の数が多いと，モデルの解釈性が下がるほか，パラメータ推定のコストや過学習の危険性が増大する．このような問題を一般に次元の呪いという．一方で，モデルの予測性能を向上させるには必要な情報はしっかり含めなければならないため，その特徴量をモデルに投入するかが重要なポイントとなる．

* モデルの解釈性
* 特徴量の解釈性  
  現実に何に対応するか説明できない特徴量を使ってモデルを構築すると，モデルを使って現象のメカニズムを説明する際に，論理的な演繹ができなくなる．
* モデル入出力の対応関係  
  モデルの入力と出力の対応が分からない．

## バリデーションの構築

モデルの汎化性能を改善していくためには，構築時にそのモデルの汎化性能を知る方法が必要である．これに対して，一般的に，学習データを学習に用いるデータ（trainデータ）と評価用のデータ（validデータ）に分け，validデータへの予測性能を何らかの評価指標で表して評価することを考える．validデータへの分け方にはいくつかの方法があるが，ここではホールドアウト検証とK-分割交差検証の2つを紹介する．

* ホールドアウト検証（hold-out validation）  
  単純に学習データをtrainとvalidの2つに分け，trainでモデルの学習，validで予測性能評価を行う．

＜ホールドアウトの図＞

* K-分割交差検証（K-fold cross validation）  
  全体のデータをK個のブロックに分割し，そのうちK-1個のブロックをtrain，残った未使用の1個をvalidとする．Validの選び方にはK通りあるが，この全ての場合についてモデル学習・評価を行い，性能の平均値をとって最終的なモデルの性能とする．

＜K-分割交差検証の図＞

学習データとテストデータがランダムに分割されているということを前提としている．

## モデル種類の選択

当日の分析課題で候補とする予定の4モデルを紹介する．なお，ここでは，概要と使用時のポイントのみを述べるが，より深く学習したい方は参考文献●等を参照されたい．

### ロジスティック回帰

・相関の高い変数がある場合は、一方を説明変数から外すなどの工夫が必要

・説明変数と事象の起こりやすさの関係が非線形の場合は、変数加工により線形化する工夫が必要

・外れ値も引きずられて悪影響を及ぼすので、丸める必要あり。

### 決定木

本来，回帰にも分類にも使えるモデルだが，ここでは分類を例に解説する．

決定木は条件分岐によって与えられたデータがどのクラスに分類されるのかを決めるアルゴリズムである．

* 結果の解釈がしやすい
* 過学習しやすい

### ランダムフォレスト

過学習しやすいという決定木の欠点を補ったのがランダムフォレストである．フォレスト（森）という名の通り，決定木を複数生成して，それらの多数決で分類結果を予測するというものである．複数の決定木を生成するために，元のデータセットからランダムにサンプルを取り出し，それについて決定木を作るという手続きを行う．

なお，このように複数のモデルを組み合わせることをアンサンブル学習という．組み合わされるモデルのことを弱学習器，最終的に出来上がったモデルのこと強学習器という．

### ニューラルネットワーク

非線形で複雑な形のモデルを実現する方法の1つが，ニューラルネットワークである．ニューラルネットワークは脳の仕組みをもとに開発されたモデルで，単純な計算を行う要素（ノード）をネットワーク状に組み合わせることで予測値を出力する．

下図はニューラルネットワークの中でも基本的なもので，順伝播型ニューラルネットワークと呼ばれる．ここではデータを入力するノードたちの層を入力層，予測値が出力される層を出力層，間に挟まれた層を中間層という．

個々のノード（パーセプトロンともいう）では，矢印で繋がっているノードから値を受け取り，それに重みを掛け合わせてから全て足し上げている．そして，さらにこの値を活性化関数という関数に代入する．活性化関数は，受け取った全体の入力値をどの程度次のノードに繋げるかを表現する非線形関数である．活性化関数には様々なものがあり，目的やデータに応じて選択する必要がある．そうして計算された値を矢印で繋がった次のノードに渡すようになっている．

《ニューラルネットワークの図》

《活性化関数の例》

特徴

* 非線形性や変数間の相互作用が反映される

注意点

* 特徴量は数値
* 計算の仕組み上，欠損値を扱うことができない
* 基本的には，特徴量を正規化などでスケーリングする必要がある

## モデルの評価

２値分類の評価指標は，大別すると以下2つに分けられる．

### 混同行列（confusion matrix）

評価指標ではないが，正例か負例であるかを予測値とする評価指標で利用されるため説明する．予測値と秦の値の組み合わせは，予測値を正例としたか負例としたか，その予測が正しいか誤りかによって，以下4つに分けられる．

* TP（True Positive，真陽性） ：予測値を正例として，その予測が正しい場合
* TN（True Negative，真陰性） ：予測値を負例として，その予測が正しい場合
* FP（False Positive，偽陽性） ：予測値を正例として，その予測が誤りの場合
* FN（False Negative，偽陰性） ：予測値を負例として，その予測が誤りの場合

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 真の値 | |
| 正例 | 負例 |
| 予測値 | 正例 | 真陽性  TP | 偽陽性  FP |
| 負例 | 偽陰性  FN | 真陽性  TN |

### accuracy（正答率）とerror rate（誤答率）

accuracyは予測が正しい割合，error rateは誤っている割合を示す．accuracyは正解したデータ数を全データ数で割ることで求められる．以下は混同行列の要素を用いた式である．

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

以下のようにscikit-learnのmetricsモジュールのaccuracy\_scoreを用いて計算できる．

《pythonコードの例》

### precision（適合率）とrecall（再現率）

presicionは正例と予測したもののうち真の値も正例の割合，recallは真の値が正例のもののうちどの程度を正例の予測として含めることができているかの割合である．混同行列の要素を用いて以下の式で表される．

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

それぞれ0から1の値をとり，1に近づくほど良いスコアである．precisionとrecallは，互いにトレードオフの関係になっている．（つまり，片方が高くなると一方は低くなる．）したがって，片方を無視すれば，もう片方を1に近づけることができるため，単体として使うことは少ない．

したがって，分析目的に応じて優先度を付けて確認すると良い．誤検知を少なくしたい場合はprecisionを重視し，正例の見逃しを避けたい場合はrecallを重視することになる．また，後述のF1-scoreはこれらのトレードオフを考慮した指標になっている．

以下のようにscikit-learnのmetricsモジュールのprecision\_score，recall\_scoreを用いて計算できる．

《pythonコードの例》

### F1-score（F値）

F1-scoreは前述のprecisionとrecallの調和平均で計算される指標である．Precisionとrecallのバランスをとった指標となっている．

|  |  |  |
| --- | --- | --- |
|  |  |  |

### logloss

正例である確率を予測値とする場合の使える指標としてloglossを紹介する．cross entropyと呼ばれることもある．

|  |  |  |
| --- | --- | --- |
|  |  |  |

ここで，は正例かどうかを表すラベル（正例が1，負例が0）を，は各レコードが正例である予測確率を表す．

は真の値を予測している確率で，真の値が正例の場合は，負例の場合はである．

loglossは，上式の通り，真の値を予測している確率の対数をとり，符号を反転させた値である．そのため，低い方が良い値であると言える．

《loglossの計算例》

以下のようにscikit-learnのmetricsモジュールのlog\_lossを用いて計算できる．

《pythonコードの例》

### AUC

AUCはROC曲線（Receiver Operating Characteristic Curve）が描く曲線をもとに計算される．ROC曲線は，予測値を正例とする閾値を1から0に動かし，そのときの偽陽性率と真陽性率をプロットして描いたものである．このROC曲線の下部の面積がAUCである．

## ハイパーパラメータのチューニング

## モデルの解釈

# メカニズムの理解

データをうまく説明するモデルはデータの生成過程を捉えている，という前提のもとで，データの生成のモデルを調べる．一般にモデルの構成要素が，変数・数理構造・パラメータであることを踏まえると，以下のような方針が考えられる．

* 数理構造から説明する
* 推定したパラメータから説明する
* 数理モデルのパラメータを変化させた状況をシミュレーションする