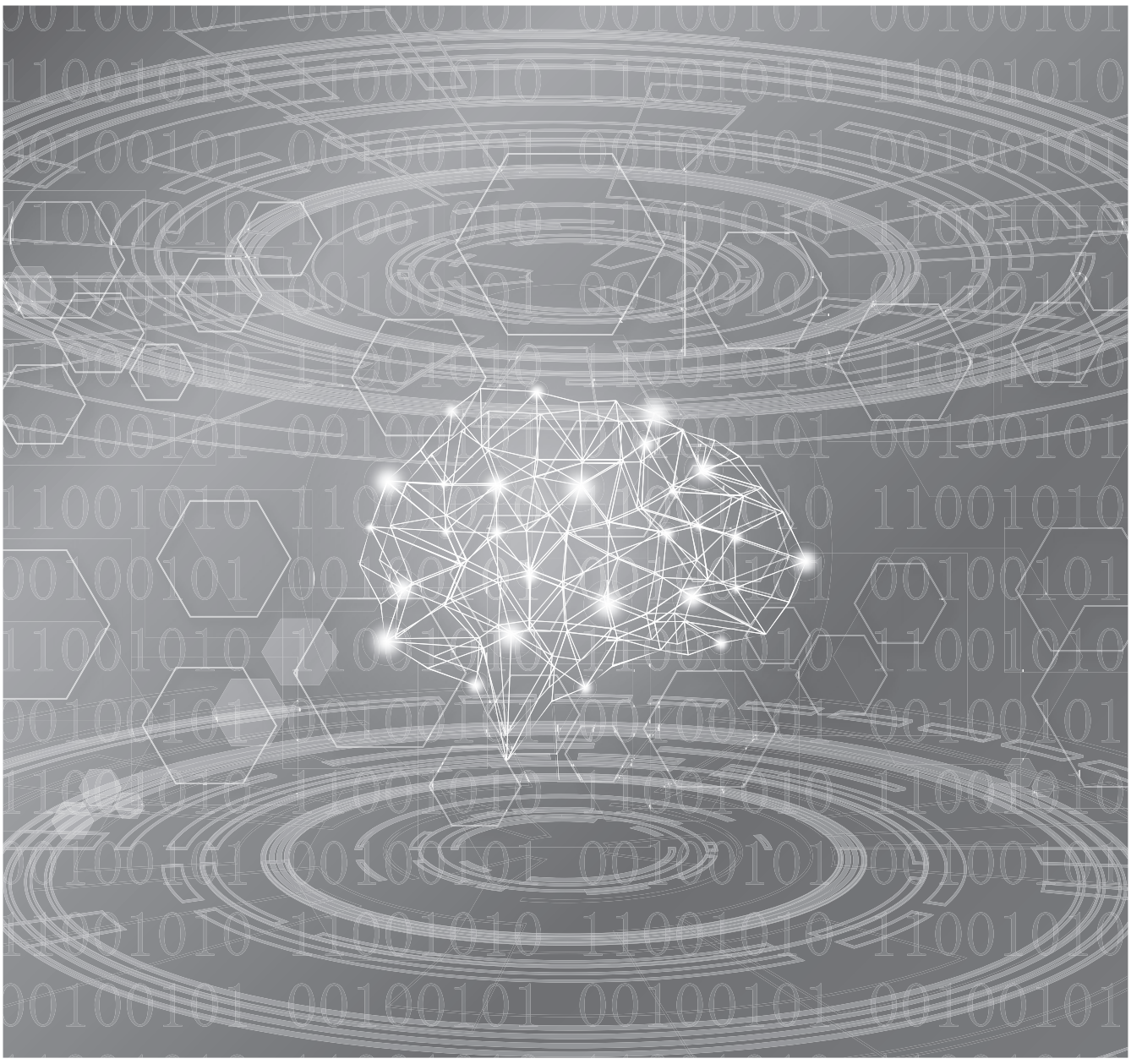
2019年度「専修学校による地域産業中核的人材養成事業」

教員用研修プログラム



Society5.0実現のためのIT技術者養成モデルカリキュラム開発と実証事業

2019年度「専修学校による地域産業中核的人材養成事業」

教員用研修プログラム

Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

AI・機械学習・データマイニング教員育成研修

■前提要件

○機械学習について

技術要件

・mac/windows の操作をしたことがある。mac 操作の経験があることが好ましい

・mac におけるターミナルの操作、windows におけるコマンドプロンプトの操作ができる

・python でファイル操作、四則演算など簡単な処理が実装できること

数学の知識

・数に関する基本事項（算術演算、べき乗、階乗、約数・倍数、数の偶奇、素数、など）を学習していること

・合計、平均、標準偏差など基本的な統計処理を理解していること

言葉に対する知識

・データサイエンス、ビッグデータ、機械学習、AI など、データ分析に関わる種々のキーワードを知っていること

○データマイニングについて

※上記「機械学習」の要件は満たしている前提で、以下のスキルも必要となります

技術要件

・scikit-learn などのライブラリを駆使した python プログラミングができること

・何らかの機械学習器を実装したことがあること

・データセットを学習データとテストデータに分け、学習データでモデルを作成し、テストデータで精度を確認するなど、基本的な機械学習のプロセスを遂行できること。

・プログラミング実行中にエラーが発生した際に、web の情報などを元に自身でエラーを解消できること。

数学の知識

・行列計算を理解していること。

言葉に対する知識

・分類、回帰、クラスタリング、次元削減など基本的な用語を知っていること。

■研修：

スケジュール	概要	説明
10：00	オリエンテーション	・本プログラムの趣旨説明
10：30	データマイニング導入 Python の基本操作 Python の応用	・現場でシステムができるまでの過程において、データマイニングはどのような役割を担うか ・Python における関数の使い方 ・Python による統計処理、機械学習の導入
12：00		
13：00 18：00	機械学習（１） 機械学習（２） 機械学習（３）	・分類問題の取り扱い ・回帰問題の取り扱い ・主成分分析等、よく使うアルゴリズムの説明

目 次

データマイニングの概要

アジェンダ

- データマイニングとは
- データマイニングの手法の分類
  - クラス分類
  - 回帰
  - クラスタリング
  - パターン抽出
  - その他の手法
- データマイニングの歴史と発展

全15回の講義について

- データマイニングの各種アルゴリズムの理解を目標とする。
  - プログラミング言語としてはPython
  - Pythonの各種ライブラリを利用してデータ分析に必要なスキルの習得を目指す
  - 第2回以降の講義で詳細を取り扱う

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

事例：マーケット・バスケット分析

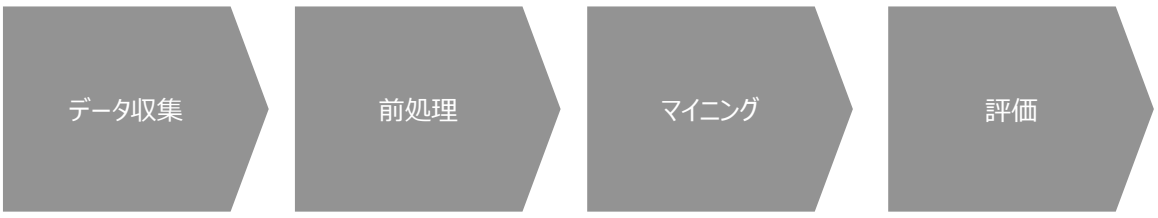
- データマイニングで有名な事例として、マーケット・バスケット分析があります。
- マーケット・バスケット分析とは、データ同士の関係性を分析するもので、どの商品とどの商品をどのような顧客が同時に購入したかを分析する手法です。

- 夕刻、紙おむつとビールが同時に購入される、という有名な事例がアメリカにあります。夕食の準備に忙しい母親に言われて商店に紙おむつを買いに来た父親が、自分へのご褒美にビールを買うため、と解釈されています。

データマイニングのプロセス

- データマイニングを行うために、まずはデータを収集することが必要です。一般的には、元となるデータが多ければ多いほど、有益な情報を採掘（マイニング）できる可能性が高まります。

- 収集されたデータは、データマイニングの各種アルゴリズムに適した形式に変換する「前処理」が施されます。



データマイニングのステップ

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

データマイニングの歴史

年号	できごと
1960年代	メインフレームが金融企業の基幹業務システムとして稼働開始した。同時に、デジタルデータの収集,蓄積,利用の試みが開始された。
1970年代	1971年から1973年にかけて、チリでサイバーシン計画が実行される。コントロールセンターが、テレックスを介して実時間でチリ各地に点在する工場からデータを収集して、収集したデータを元に、オペレーションズ・リサーチを用いて最適化した生産計画を作成し、工場に対して生産計画をフィードバックするシステムであった。
1980年代	現在の"Data mining"の定義と類似する"Knowledge Discovery in Databases"という語が出現する。関係データベースシステムとその操作用言語であるSQLが出現する。データウェアハウスの運用が開始される。
1990年代	1990年頃から始まった計算機の急激な性能向上により"Knowledge Discovery in Databases"の研究が大幅に加速される。 1999年 - 2010年代に大量の実世界データを収集・供給する基盤となるInternet of Things(IoT)の用語がKevin Ashtonにより初めて使用された。(この当時のIoTは、様々な物体にRFIDタグを貼り付け、RFIDに対応したセンサーを用いて物体からの情報収集を行い、収集した情報を活用することを指していた)
2000年代	インターネットへの常時接続が一般家庭にも普及する。インターネット上に蓄積されたデータが加速度的に増加する。後にデータの主要な供給源の1つとなる友人紹介型のソーシャル・ネットワーキング・サービスが2002年より相次いで提供され始める。コンピュータとインターネットの普及に着目し、ビジネスにおいて膨大に蓄積され活用しきれなくなったデータの分析を専門に行う企業も徐々に出現し始める。
2010年代	英国"The Economist"誌において"big data"の語が提唱された。コモディティ化によりコンピュータの計算能力が安価になり、高速データ処理用のコンピュータ・クラスタの構築が容易にできるようになった。データ分析のコストが下がり、ビッグデータ解析の応用が進むようになった。データサイエンティストという名称の職業が台頭し始めた。また、ビッグデータを用いたデータマイニングを応用したサービスが一般向けにも提供され始めた。コグニティブ・コンピューティング・システムが商用で実用化された。テレビ番組の紹介コーナーでも、インターネット上に存在するビッグデータの統計分析結果を元に流行のトレンドを紹介するようになった。 ディープラーニングの実用化が急速に進み、非常に多数の人工知能サービスが現れた。

出展： https://ja.wikipedia.org/wiki/データマイニング

データマイニングに用いられるツール・ライブラリ
R言語

Wikipediaより（https://ja.wikipedia.org/wiki/R言語）

- R言語（あーるげんご）はオープンソース・フリーソフトウェアの統計解析向けのプログラミング言語及びその開発実行環境である。ファイル名拡張子は.r, .R, .RData, .rds, .rda。
- R言語はニュージーランドのオークランド大学のRoss IhakaとRobert Clifford Gentlemanにより作られた。現在ではR Development Core Team[注 1] によりメンテナンスと拡張がなされている。

データマイニングに用いられるツール・ライブラリ
Python + Jupyter notebook + scikit-learnなど

Wikipediaより（https://ja.wikipedia.org/wiki/Scikit-learn）

- scikit-learn (旧称：scikits.learn) はPythonのオープンソース機械学習ライブラリ[2]である。サポートベクターマシン、ランダムフォレスト、Gradient Boosting、k近傍法、DBSCANなどを含む様々な分類、回帰、クラスタリングアルゴリズムを備えており、Pythonの数値計算ライブラリのNumPyとSciPyとやり取りするよう設計されている。

データマイニングに用いられるツール・ライブラリ
Weka

Wikipediaより（https://ja.wikipedia.org/wiki/Weka）

- Weka (Waikato Environment for Knowledge Analysis) は、ニュージーランドのワイカト大学で開発した機械学習ソフトウェアで、Javaで書かれている。GNU General Public License でライセンスされているフリーソフトウェアである。

Python基礎

Pythonの基礎

アジェンダ

- Pythonの基礎文法
  - 四則演算
  - 文字列
  - 変数
  - リスト
  - 制御構文
    - if文による条件分岐処理
    - for文によるループ処理
  - 関数
- 演習課題

四則演算

- 電卓の用に基本的な演算を実行可能
  - 足し算
  - 引き算
  - 掛け算
  - 割り算
- 累乗の計算などもできるため、通常の電卓よりもリッチな計算ができる

```
2 + 2
>>> 4

2 * 5
>>> 10
```

文字列

- シングルクォート、もしくはダブルクォートで囲うことで文字列にできる
- 日本語も扱える

```
“hoge”
>>> “hoge”

‘日本語もOK’
>>> 日本語もOK
```

print関数

- 変数の内容や計算結果を表示するための便利な関数

```
print(2+2)
>>> 4

print(‘hoge’)
>>> hoge
```

変数

- 様々な値が代入可能な箱のようなもの
- CやJavaのように事前に変数の型を指定する必要はない
- 変数の内容はprint関数で表示することができる
- 変数は再代入も可能

```
a = 1
b = ‘hoge’

a = ‘fuga’
```

文字列再び

- 文字列に対する演算
  - ‘+’演算で文字列の連結
  - ‘*’演算で繰り返し
- インデクスでのアクセス
  - 文字単位でのアクセスが可能
  - インデクスは0番から始まる
  - インデクスに負の値を入れると文字列の終端からのアクセスとなる
  - 範囲指定で部分文字列を取り出せる
- immutableな変数
  - 部分文字列の変更は不可能
  - インデクスでアクセスし、再代入しようとするとエラーとなる

リスト

- 複数の値をまとめて保持することのできるデータ構造
  - CやJavaの配列のようなデータ構造
  - 数値でも文字列でも保持可能で、1つのリストに数値と文字列を混ぜて保持することも可能
- 文字列と同じようにインデクスによるアクセスが可能
  - 文字列と異なる点はmutableであるため、再代入が可能である点

```
nums = [1, 2, 3, 4, 5]

words = [‘hoge’, ‘fuga’]
```

制御構文

- プログラムのバリエーションを増やす上で必須の構文
  - if文による条件分岐
  - for文によるループ

if文による条件分岐

- ifの後に条件式を書く
  - 評価した結果、条件に該当する場合は処理が動く
    - CやJavaと同じ
  - 一方、CやJavaとは異なり、インデントで揃えたコードブロック単位で条件節が決まる
- ifに該当しない場合、必ず通る条件として、elseを使うことができる
  - CやJavaと同じ
- elifを利用して、複数の条件判定を行うことも可能

```
if x == 50:
        print(‘50です’)
else:
        print(‘50ではありません’)
```

for文による繰り返し

- Pythonのfor文はリストを走査してループさせるのが基本
  - リストの要素を取り出しながら処理を進めていく
- CやJavaのforループのような処理はできないのか？
  - ループの変数を任意の数までインクリメントしながらループ処理の実行
  - range関数を使えば実現可能
- 多重ループ
  - ループ内でループをまわすことができる

```
nums = [1, 2, 3, 4, 5]
for num in nums:
        print(num)
```

break文とcontinue文

- break文
  - その時点でループを打ち切る
  - 特定の条件を満たした時は後続のループ自体を処理したくない時に利用
- continue文
  - その時点でループの先頭に戻る
  - 特定の条件を満たした時はループ内の後続の処理をしたくない時に利用

関数

- 処理をひとまとめにしたもの
  - 外からは任意で引数を渡すことが可能
  - 処理の結果は必ず返さなくても良い
  - 複数の結果を返すこともできる
- うまく使えばプログラムの見通しが良くなり、再利用性が高まる

関数の定義

- defキーワードで関数を定義
  - 変数のスコープは関数内で閉じる
  - そのため、同名の変数を複数の関数で利用することができる

```
def add(x, y):
        return x + y

add(1,2)
>>> 3
```

演習問題1

- FizzBuzz
  - 1から100の数字を表示する関数FizzBuzzを定義しなさい
  - - 但し、FizzBuzz関数は3の倍数の時は数字のかわりに"Fizz"、5の倍数の時は数字のかわりに"Buzz", 3と5の倍数の時は"FizzBuzz"と表示しなさい
  - - FizzBuzz関数を実行し、期待通りの挙動になっていることを示すこと

演習問題2

- 九九の表示
  - 九九を表示するプログラムを書きなさい
  - 但し、一行に一つの段の結果を全て表示すること（9x9の九九の表が表示される）

演習問題3

- フィボナッチ数列
  - 10個のフィボナッチ数を求めて表示するプログラムを書きなさい
  - フィボナッチ数とは、前の２つの数字を加えると次の数になる、数列です。（ただし、1番目と2番目は1）
  - つまり、1, 1, 2, 3, 5, 8, 13 ....のように表示するプログラム

演習問題4

- フィボナッチ数列2
  - フィボナッチ数列を関数化して、任意の個数のフィボナッチ数を返すように修正しなさい

Python応用

Pythonの応用を学ぶ

アジェンダ

- Pythonの応用文法
  - リスト型
  - タプル型
  - 集合型
  - 辞書型
  - ファイルの入出力
  - モジュール

リスト型

リストは非常に重要なデータ型の１つ

- リストを操作するための様々な関数の紹介
- リストを使ったデータ構造の実現（キュー）
- リスト内包表記

```
>>> nums = [1, 2, 3, 4, 5]
>>> nums
[1, 2, 3, 4, 5]
```

リストを操作するための便利関数群

- リストの末尾に要素を追加：append()
- リストの要素で該当する最初の要素を削除：remove()
- リストの指定した位置に要素を挿入：insert()
- リストの順序を逆順に変更：reverse()
- リストの要素をソート：sort()

その他リストに対する様々な処理が関数として提供されている。

リスト内包表記

- シーケンス要素に対してある操作を行った結果をリスト化して返すリスト生成の特別な記法
- 非常に柔軟でかつ強力なため、覚えておく

```
# リスト内包表記によって0から5までの数字を生成
>>> nums = [i for range(6)]
>>> nums
[0, 1, 2, 3, 4, 5]
```

タプル型

- リストに似たデータ構造
  - スライスを利用した要素へのアクセスができる
- リストとの違いとして、immutableなデータ構造

```
# リストのように扱えるが、再代入などはできない
>>> nums = (1, 2, 3, 4, 5)
>>> nums[1]
2
>>> nums[1] = 0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

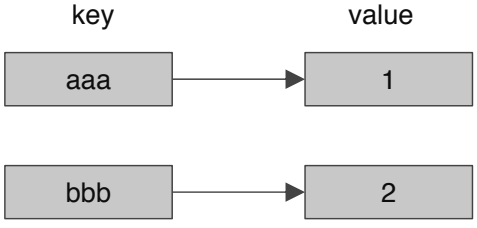
集合型

- 重複がなく、順序付けられていない要素をひとまとめにしたもの
- 様々な集合演算が扱える

```
>>> box = {'aaa', 'bbb', 'ccc', 'ddd'}
# 順序がないため、代入の時の順番が保証されているわけではない
>>> box
{'bbb', 'ddd', 'ccc', 'aaa'}
```

辞書型

- 辞書型は多言語では「連想配列」や「ハッシュ」と呼ばれているものに相当するデータ型
- キー(Key)と値(Value)の組み合わせを使うデータ型
  - Pythonのデータ構造のんかでも重要なものとなる



辞書型の基本

```
# 例えば、本のタイトルと値段とする
>>> book = {'aaa': 500, 'bbb': 1000, 'ccc': 10000}
>>> book
{'bbb': 1000, 'ccc': 10000, 'aaa': 500}
>>> book['aaa']
500

# 新しいkey-valueペアの登録
>>> book['ddd'] = 50
>>> book
{'bbb': 1000, 'ddd': 50, 'ccc': 10000, 'aaa': 500}
```

ループ処理との組み合わせ

- ループ処理に組み合わせると、key, valueのペアを順次処理できる

```
>>> book = {'aaa': 100, 'bbb': 200, 'ccc': 300}
# items()という関数でkey-valueペアを返す
>>> for key, value in book.items():
...     key, value
...
('bbb', 200)
('ccc', 300)
('aaa', 100)
```

ファイルの入出力：共通の流れ

ファイル処理の一連の流れ

- ファイルオブジェクトを作成
  - この時、モードを指定する
- モードに応じたメソッドを使ってファイルオブジェクトを経由した処理を行う
- 処理が終わったらファイルオブジェクトを閉じる

ファイルの入出力：書き込み

```
#open()にょりファイルを開きファイルオブジェクトを取得する
# 'w'は書き込みモード
# 'r'は読み出し専用モード
>>> f = open('./tmp', 'w')

# writeメソッドで書き込む
>>> f.write("This is a pen.¥n")
>>> f.write("That is a cup of coffee.¥n")

# ファイルオブジェクトを閉じる
>>> f.close()
```

ファイルの入出力：読み出し

```
# 先程書き込んだファイルを再びopen()で開く
f = open('./tmp', 'r')

# ループで一行ずつ読み出せる
for line in f:
    print(line)

# 処理が終わったらファイルを閉じる
f.close()
```

モジュールとは？

- モジュールとは、再利用可能なプログラムの部品のようなもの
- Pythonには標準モジュールとして、各種の便利な機能が提供されている
  - 数学関連
  - 日付、時間関連
  - 文字列のフォーマッティング関連
  - etc...

モジュールのインポート

```
# import文を使ってモジュールをロードする
>>> import math

# dir()を使うとインポートしたモジュールで
# 利用可能な関数や定数の名前がわかる
>>> dir(math)
# すごく色々表示されるので省略

>>> math.pi
3.141592653589793
```

演習問題1

aからbまでの整数値がランダムで要素として格納されているリストを作成する関数generate_random_list()を作成しなさい。なお、以下の条件を満たすこと。

- 引数にリストの長さを与えることができる
- ランダムな値を生成する方法はいくつかある
    - random.randint(a,b)は a <= n <= b である整数nを返す

```
>>> import random
>>> random.randint(0,5)
0
>>> random.randint(0,5)
3
```

演習問題2

問題1で作成した関数を利用して、要素数100で0から10までの整数値が要素となっているリストを作成しなさい。そのリスト内の要素をチェックし、各整数値が何回ずつ出現したかをカウントする関数を作成しなさい

- 関数の引数は、問題1で作成した関数によって生成される要素がランダムなリスト
- ランダムなリストは、要素数100で、要素は0から10の整数値
- keyにリストの要素である整数値、valueにリストの要素が何回出現したかの値を持つ辞書が返り値

統計解析全体像

統計解析の全体像を学び
この後の数回にわたる講義のガイドとする

アジェンダ

- 統計解析とは？
- データから「事実を正しく語る」ことについて
- 統計という道具を使う
  - データ収集
  - 記述統計
  - 推定
  - 仮説検定
  - 相関
  - 回帰
  - etc...

キーワード２：統計

Wikipediaの「統計」によると…

- 統計（とうけい、statistic）は、現象を調査することによって数量で把握すること、または、調査によって得られた数量データ(統計量)のことである。

第1回より再掲

つまり、統計とは・・・

現象をデータから読み解くための

技術やその結果そのもの

統計学はその技術を

体系立てた学問

なぜ統計が必要か？

- データは簡単に、かつたくさん手に入るようになった

- 次に必要なのは、データから「何が起きているのか」を正確に理解し、「次の行動の意思決定材料」とする方法
- 手元にあるデータを正しく理解するために道具としての統計が必要となってくる

統計学について

Wikipediaの「統計学」によると...

- 統計学は、経験的に得られたバラツキのあるデータから、応用数学の手法を用いて数値上の性質や規則性あるいは不規則性を見いだす。統計的手法は、実験計画、データの要約や解釈を行う上での根拠を提供する学問であり、幅広い分野で応用されている。 現在では、医学（疫学、EBM）、薬学、経済学、社会学、心理学、言語学など、自然科学・社会科学・人文科学の実証分析を伴う分野について、必須の学問となっている。また、統計学は哲学の一分科である科学哲学においても重要なひとつのトピックスになっている。

つまり、統計学とは・・・

データを正しく解釈し、意思決定を

行うために必須の学問

これまでは学問領域での適用に

とどまってきたが、ビジネス活動にも

必要とされるようになり始めている

統計解析とは？

- データサイエンティスト、ビッグデータ、統計、機械学習、AIなど、データ分析に関わる種々のキーワードを中心に説明を行い、業界イメージをつけるとともに、講義全体（全15回）の全体像を理解する
- また、講義内容をもとにして自由課題の演習を実施する

データから「事実を正しく語る」とは？

- ある事象に対して何らかの主張をしたい場合、どうすれば「正しく」主張することができるだろうか？

事象：第一子は第二子よりも身体的に大きな子に育ちやすい

もちろん、医学的に立証された正しい説ではありません。思考実験です。

事象に対しての主張方法

- 「私には子供が2人いて、第一子よりも第二子の方が大きい。なのでこの主張は正しい」「また、近所の家の子も同じである。この説は信憑性が高い」
  - これは本当に正しいと言える？

- 「私の姉には子供が3人いて、第二子が一番体が大きい。なので、この主張は正しくない」
  - この反証も正しいといえるだろうか？

この主張が正しいと言い切れない理由

- 標本数が小さすぎる
  - 「標本」とは観測されたデータのことを言う
  - そもそも標本数1や2では「たまたま」の可能性を排除しきれない
- 選択バイアス
  - そのそもこういった議論に参加する人は、この説に興味を持っている人
  - なので、主張に当てはまる事象だけを意図的に集めている可能性がある
- 確証バイアス
  - この主張が「正しい/正しくない」という前提に立っているため、公平な主張ではない
- 不正確さ
  - そもそも定性的であり「何を持って身体が大きい」かの定義がない状態で主張が行われている

統計という道具を使う

- そこで、データから客観的な事実を語るための道具として、統計を使う
- また、さらに深掘りするために統計解析を行う

- 統計解析は以下のような要素からなる
  - データ収集
  - 記述統計
  - 推定
  - 仮説検定
  - 探索的なデータ解析
    - 外れ値チェック
    - 相関チェック
    - 回帰
    - etc...

データ収集

- まずは分析に必要なデータを集める
  - データがないと分析自体が行えない
- 良質なデータを多く集めることが重要
  - が、実際は良質なデータほど多くのデータを集めることは難しい

- 以後、Pythonのscikit-learnに同梱されているサンプルデータであるirisを例にして話を進めて行く

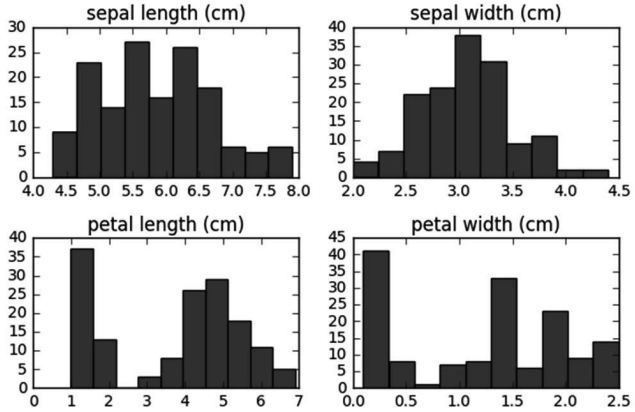
記述統計

- データを集計しその特性について簡潔に理解しやすくする方法

- ヒストグラム（度数分布）
- 平均値、中央値
- 分散、標準偏差
- 四分位数
- etc..

ヒストグラム（度数分布表）

- データのばらつきを確認するための方法
- １つの変数毎に値を見ていく



irisデータの場合、変数が4つあるため、それぞれでヒストグラムを確認

変数毎にデータのばらつきが異なることが分かる

平均値、中央値、分散、標準偏差、四分位数

- pandasの関数で各変数について一気に確認することができる

・レコード数
・平均値
・標準偏差
・最小値
・四分位数(25%)
・中央値
・四分位数(75%)
・最大値

の順で並んでいる

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

平均値の意味

- 平均値
  - データの総和をデータの個数で割ったもの
  - データ１つあたりの数値を表す

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

nはデータの個数：irisの場合は150
xは1つ１つのデータの値：irisの場合、１つのデータのとある変数

数式の出典：https://ja.wikipedia.org/wiki/%E5%B9%B3%E5%9D%87

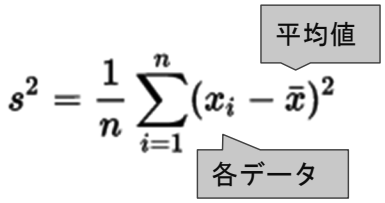
中央値の意味

- 値を小さいものから順に並べ直す
- ちょうど真ん中の値が中央値となる
  - irisの場合だと、データ数が150個のため、小さいもの順に並べた75番目の数字

- 平均値と中央値の関係
  - 平均値の意味は「代表的によく現れる数値」とも言える
  - 代表的によく現れる数値が「真ん中」に表れている場合、平均値と中央値は近しい値となる
  - 逆に言うと、平均値は必ず「真ん中」の数字ではないことに注意する必要がある

分散

- 分散
  - データがどの程度散らばっているか、を表したもの
  - 平均値と各データの差の二乗を、全てのデータについて足し合わせたもの



二乗を取る理由としては、平均値より大きい値は正、小さい値は負になってしまい、打ち消し合う可能性があるため。
二乗を取ることで必ず正の値とし、尺度として使えるようにする
※意図的に二乗をとる数学的なテクニック

数式の出典：https://ja.wikipedia.org/wiki/%E5%88%86%E6%95%A3

標準偏差

- 分散に対して平方根を取ったもの
  - 分散は二乗しているため、もとのデータと単位が異なる
    - 分散を見ているだけでは、実際のデータではどういう意味を持つのかがイメージしにくい
  - 平方根を取ることで、元のデータと単位をそろえた、ばらつきの尺度

- 直感的に理解されやすいという観点で、ばらつきの尺度としては、標準偏差の方が採用されやすい

母集団、標本

- 母集団
  - 調査対象になっているデータの全体を表す
- 標本
  - 調査対象になっているデータから、実際に取得されて手元にあるデータ

- irisを例にすると
  - 世の中にある三種類のアヤメ（Setosa, Versicolor, Virginica）全体が母集団
  - 実際に全てのデータを入手することは不可能なためサンプルとして一部のデータしか得られない。それが標本
- 標本から母集団の性質を明らかにすることが統計解析の大きな目的の１つ

演習問題

- 身のまわりで平均値と中央値が一致しないような例を探してみよ
  - なぜ平均値と中央値が一致しないのかその理由を述べよ

相関

相関について学ぶ

アジェンダ

- 相関について
- 相関関係と因果関係
- 2変数の相関関係
  - 相関係数
  - 散布図によるプロット
- 2変数以上の相関関係
  - 相関行列

相関について

- 2つの変数の間の関係性について数値で記述した分析手法
- 一方が変化すればそれに応じて他方も変化する、という関係
  - Aが上がれば、Bも上がる
  - Bが上がれば、Aも上がる

- 似ている概念に因果関係があり、それとは異なるので注意が必要
  - 因果関係もAが上がれば、それが原因でBが上がる、という関係
  - ただし、Bが上がればAが上がるとなはならないため、一方向の関係

2変数の相関関係：相関係数

- 相関係数は、2つの変数に相関があるかないか、またその強さについて表した指標
- -1から1までの値を取る
  - 0：無相関
    - そもそも２つの変数に相関関係はない
  - 0から1：正の相関
    - 2つの変数に正の相関がある。正の相関とは、片方の変数が大きくなるともう一方の変数も大きくなる、ということ
  - 0から-1：負の相関
    - 2つの変数に負の相関がある。負の相関とは、片方の変数が大きくなるともう一方の変数は小さくなる、ということ

相関係数の求め方

- (相関係数) = (共分散) ÷ (変数Aの標準偏差 × 変数Bの標準偏差)
- 共分散は、下記の値を全ての観測値で足し合わして観測数で割ったもの
  - (変数Aのとある観測値 - 変数Aの平均)×(変数Bのとある観測値 - 変数Bの平均)

- すなわち、以下の情報があれば相関係数を求めることができる
  - 変数Aの平均と標準偏差
  - 変数Bの平均と標準偏差

相関係数を求めるの例題

- irisのsepal length(cm)（がく片の長さ）とpetal length(cm)（花弁の長さ）の相関係数を求めよ

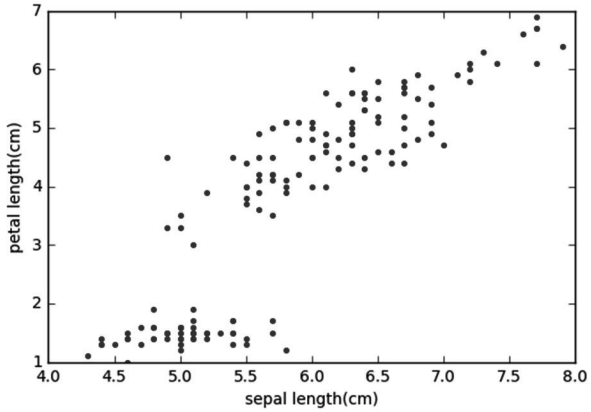
解答

- 共分散は1.27
- 相関係数は0.87
  - 非常に強い正の相関があるといえる

- なお、Pandasには変数間の共分散、相関係数を求めてくれる関数も用意されている
  - cov()関数
  - corr()関数

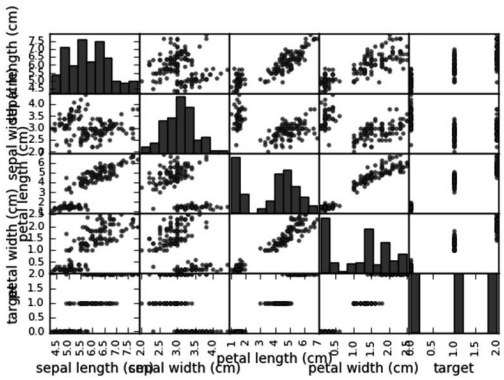
2変数の相関関係：散布図によるプロット

- irisの’sepal length(cm)’, ‘petal length(cm)’ の関係を散布図としてプロット
- 正の相関が強い変数のため、右肩上がりとなっている
  - 逆に負の相関だと右肩下がり、無相関だと方向性がなくなる



2変数以上の相関関係：相関行列

- 変数ごとの相関関係を一気に見るものを相関行列と呼ぶ
  - pandasには実現する関数が提供されている
  - 対角線がヒストグラム
  - それぞれの変数の掛け合わせでの散布図が描かれる



演習問題

- 例題ではirisの種類自体には着目していなかったので、種類にも注目して相関関係を調べる

- irisのSetosaとVersicolorのがく片の幅と花弁の幅の相関係数を求めよ。また散布図を図示せよ。
- irisのVersicolorとVirginicaのがく片の長さと花弁の幅の相関係数を求めよ。また散布図を図示せよ。

機械学習全体像

アジェンダ

- 機械学習とは？
- 機械学習で使われる用語の整理
- 機械学習の処理の流れ
  - データの入手
  - データの前処理
  - 機械学習の手法選択
  - パラメータの選択
  - モデルの学習
  - モデルの評価
  - チューニング

機械学習とは？

- 人間が行っていることと同等の学習能力をコンピュータで実現しようとする研究課題の1つ
- より汎用的に言うと、データからルールから見つけ出すアルゴリズムのこと
- 事前に正解データを与えることで学習し、学習した結果から未知のデータに対しても適用できるルールを作り出す手法を教師あり学習という
- 正解データという概念がないデータに対し、データの背後に潜む構造を見つけ出すことでルールを見つける手法を教師なし学習という

機械学習で使われる用語の整理

- 特徴量
- ラベル
- モデル
- 学習
- 学習データ
- 過学習
- チューニング

特徴量

- 特徴ベクトル、などとも呼ばれる
- 物や現象の状態をデータの羅列で表現したもの
- 特徴量自体は何でも良いが、一般的には、機械学習に学習させたいことに対して特徴がよく捉えられているもののの方が良い
- 例えば、人を表現することを考える
  - 表現方法１：身長と体重の2つで表現
    - [170, 50], [175, 70], [165, 70], [180, 80], etc...
  - 表現方法2：50m走のタイム、握力、背筋力
    - [6.5, 50, 100], [7.0, 45, 150], etc...
- 特徴量の数を「データの次元」と呼ぶことも多い

ラベル

- 特徴量で表現されているものが何であるかを表現したもの
  - 主にラベルがついているデータに対して行う学習が教師あり学習と呼ばれる
- ある状態であるか、そうでないか、というラベルを付けることが多い
  - その場合、ラベルが2種類になるため、二値ラベルと呼ばれ、二値のラベルを分類する問題は二値分類問題と呼ばれる
    - 例えば、故障しているか、そうではないか、というラベル
  - 機械学習の分類問題で最も基本的な問題となる
- 複数のラベルをつけても良い
  - その場合は、複数のラベルに分類する多値分類問題と呼ばれる問題となる

学習

- データからモデルを見つけ出す処理の過程のこと
  - 見つけ出す処理の過程にも様々な考え方がり、それが機械学習の各種アルゴリズムとなっている
- 教師あり学習の場合
  - ラベルと特徴量の組み合わせから、あるラベルを表す特徴量にはどのようなルールがあるのかを機械的に見つけることに相当する
- 教師なし学習の場合
  - ラベルに関係なく、与えられた特徴量からデータの背後に潜む構造を探すことに相当する

学習データと過学習

- モデルの構築のために使われるデータのこと
- 機械学習の一般的な流れとしては、以下のとおりになる
  - 手元のデータを、学習データとテストデータ（評価データ）に分ける
  - 学習データにアルゴリズムを適用して学習を行う
  - 学習がうまくいったかの確認にテストデータを使う
- このようにデータを分割する理由として、過学習を割けることが目的となる
- 過学習とは、学習データに対してのみ高い分類精度を示すようなモデルができてしまうこと
  - つまり、学習に使ったデータに対して「のみ」うまく分類ができている状態
  - それを割けるために、評価は学習に使ったデータ以外のものを利用する

チューニング

- モデルの精度を高めるために行う作業
- 機械学習の各種手法には、パラメータが存在する
- パラメータを適切に設定することが機械学習のモデルの精度を高める重要なポイントとなる

機械学習の処理の流れ

1. データの入手
2. データの前処理（加工、整形、尺度の変換など）
3. 機械学習の手法選択
4. パラメータの選択
5. モデルの学習
6. モデルの評価
7. チューニング

機械学習で想定されているデータ

- 教師あり学習の場合は、１つの事例に対してラベルと特徴量がセットになって与えられるデータ
  - [ラベル、(特徴量)]
- 教師なし学習の場合、ラベルは不要。1つの事例に関して特徴量が与えられるデータ
  - [(特徴量)]

前処理

- 手元にあるデータを、機械学習で想定しているデータ形式に変換すること
- 単純にデータ形式を整えるだけでなく、得られている特徴量を加工して新しい特徴量を作ったりもする
  - 例：身長と体重から、BMIの値を作り新しい特徴量にする
- その他、異常値の取扱いを決めたりもする
- 機械学習はデータの整備が一番重要な課題のため、実務的には、前処理に大多数の時間を使う

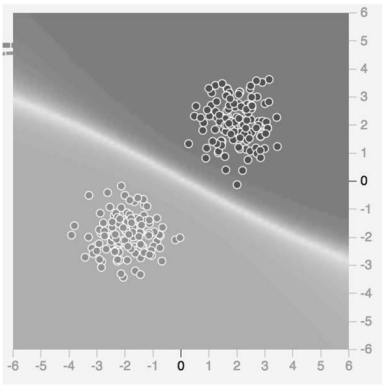
機械学習でできることの整理

- 分類
  - データからラベルを学習し、ラベルを予測する
- 回帰
  - データに当てはまるモデルを検討する
- クラスタリング
  - データの似ているもの同士をまとめて、データの構造を新しく発見するもの
- 次元削減
  - データの次元を落とすことでより本質的な情報を作ったり、可視化しやすくする

「分類」の機械学習手法

- データの集合をラベル毎に分類する線（正確には超平面）を決める

- SVM（Support Vector Machine）
- k-近傍法
- 決定木
- ランダムフォレスト
- etc...

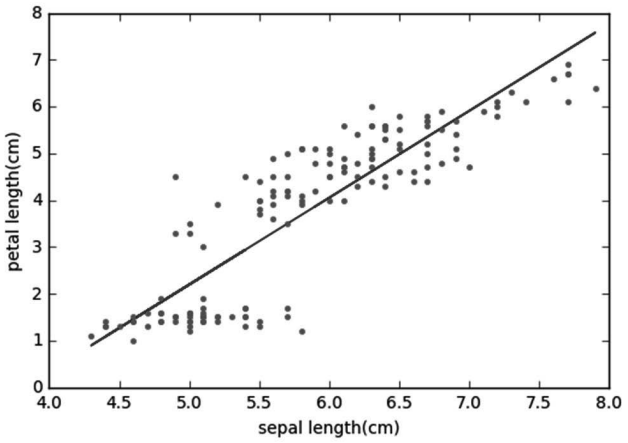


青色と橙色のラベルを分類するような線を引く

「回帰」の手法

- 統計の章で取り上げたもの。機械学習の一手法とみなすことができる
- 回帰によって「予測」ができるようになる

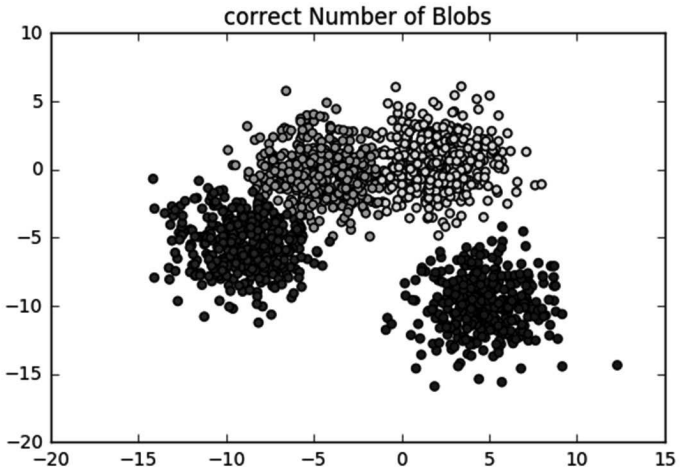
- 線形回帰
- ロジスティック回帰
- リッジ回帰
- etc...



「クラスタリング」の手法

- 教師なしの手法であるため、似通ったもの同士をおおざっぱにまとめるものと考えれば良い

- k-means（k平均法）
- 階層的クラスタリング
- 混合ガウス分布
- etc...



「次元削減」の手法

- 次元削減の主な目的は２つ
  - 特徴量を減らすことでより本質的な情報だけを残す
    - これにより場合によっては、モデルの精度は向上する
  - 可視化可能な次元まで特徴量を減らすことでデータの見える化を行う
    - データを可視化して得られることは多い

- 主成分分析
- 独立成分分析
- 非負値行列因子分解
- etc...

モデルの評価

- 学習がどれくらいうまくいったかを評価するプロセス
- 単純に分類がどれくらいうまくいったかだけに着目していると、過学習に陥ってしまう可能性があるため注意が必要
- 評価するための代表的な指標や考え方は下記の通り
  - Accuracy（精度）
  - Confusion Matrix（混同行列）
  - Precision（適合率）、Recall（再現率）、F値
  - Precision-Recall Curve
  - ROC Curve（Receiver Operating Characteristic Curve：ROC曲線）
- いずれも教師あり学習の手法で適用されるもの
  - 教師なし学習は、そもそも正解のラベルがないため、評価がしにくい

Accuracy（精度）

- 予測されたラベルが、正解ラベルに対してどれくらいあっているかという指標
- 例えば、以下の場合を考える
  - 予測されたラベル：[0, 1, 1, 1, 0]
  - 正解のラベル：[0, 1, 1, 0, 0]
  - 5個中4個正解なので、精度は4 / 5 x 100 = 80% となる
- 精度は一般的な指標であるが、精度のみを見ていると見落とす情報も多い

Confusion Matrix（混同行列）

- 真のラベルと、予測したラベルについて行列で表現したもの
- 精度だけでは見えなかった「正と予測したが実は負だったラベル」「負と予測したが実は正だったラベル」が確認できるようになる
  - 「正と予測したが実は負だったラベル」は偽陽性とも呼ばれ、問題があることを見逃してしまうことに相当するので、問題によっては割けなければいけない間違い
  - 「負と予測したが実は正だったラベル」は偽陰性とも呼ばれ、問題がないものを問題であるとしてしまうことに相当する

混同行列の例

- データ件数5件
- 正しいラベル[0, 1, 0, 1, 0]とする（0は負、1は正のラベルとする）
- 予測したラベル[0, 1, 1, 0, 0]とする

		予測したラベル		
		正	負	合計
正しいラベル	正	1	1	2
	負	1	2	3
	合計	2	3	5

Precision, Recall, F値

- Precision（適合率）とは、正と予測したラベルのうち、実際に正だったものの割合
  - すなわち、正しく予測できたものの、全体に対する割合
- Recall（再現率）とは、実際に正であるラベルのうち、実際に正と予測されたものの割合
  - すなわち、正しく予測されたものの全体に対する割合
- F値とは、適合率と再現率の調和平均
  - 適合率と再現率は一般的にトレードオフの関係にあるため、双方を同時に評価するための指標

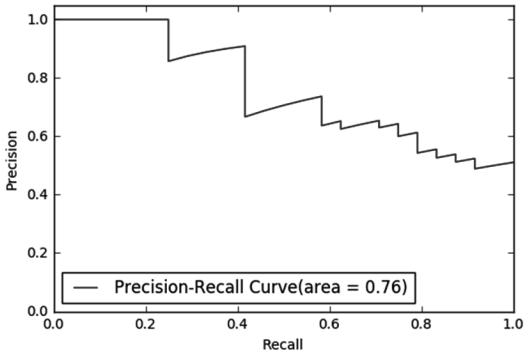
混同行列とPrecision/Recallの関係

		予測したラベル		
		正	負	合計
正しいラベル	正	1	1	2
	負	1	2	3
	合計	2	3	5

混同行列からPrecisionとRecallは算出することができる
- Precision = 1 / 2 = 0.5
- Recall = 1 / 2 = 0.5

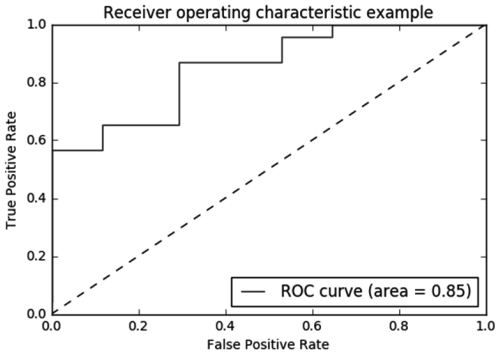
Precision-Recall Curve

- データの件数をランダムに1件から順に増やして行きながら、適合率と再現率を調べてプロットしたもの
- このグラフの下側部分をAUC（Area Under the Curve）と呼び、精度の指標として用いられる



ROC曲線

- 横軸にFalse Positive（偽陽性）の値、縦軸にTrue Positive（正しく分類できた）の値をプロットする
- Precisino-Recall Curveと同じく、データを1件からランダムに取り出し、上記の値を調べ、プロットしていったもの
- 同じく曲線下部の面積はAUCと呼ばれる



AUCについてもう少し

- Precision-Recall Curveでも、ROC曲線でもAUCは0から1までの値を取る
- 完全にきれいな分類ができているものは面積が1となる
- 一方、完全にランダムに分類した際も面積は0.5となる
  - つまり、一般的には0.5より大きな値となる
  - 0.5を下回る場合は、非常に良くない状態

演習問題

- 機械学習を利用している面白いサービスやシステムの例を探してみよ
- そのサービスには、どんなデータを使って、どんな機械学習の手法が使われていると思いますか？特に以下の観点についてまとめてみよ
  - データ
  - 機械学習の手法

決定木

アジェンダ

- 振り返り
- 決定木による発話の分類

振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

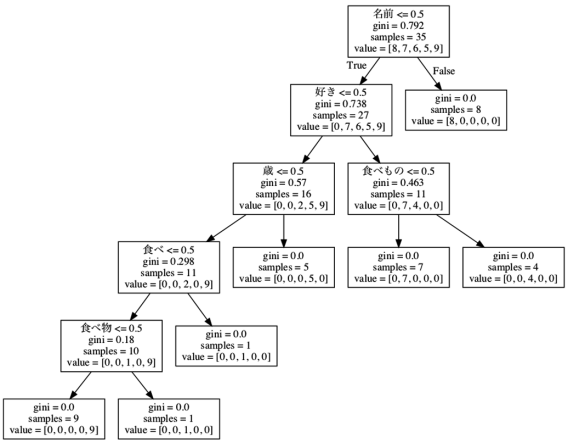
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

決定木による発話の分類

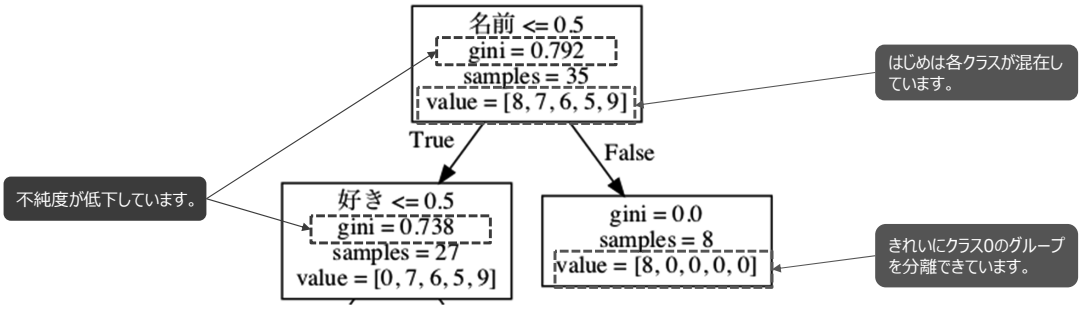
決定木とは

- 決定木分析は回帰や分類を実行する手法です。
- 顧客情報やアンケート結果などの目的変数に寄与する説明変数を見つけ、樹木状のモデルを作成します。



決定木の分岐の指標

- 決定木においてデータを分割する基準は「情報利得」と「不純度」です。
- 情報利得とは、いかにうまく分割できたか（分割前の不純度 - 分割後の不純度）を表します。
- 不純度とは、分割後のグループにおける、複数のクラスの混在の度合い（どれだけごちゃごちゃしているか）を表します。



演習：Decision Tree（決定木）による対話分類モデルの構築

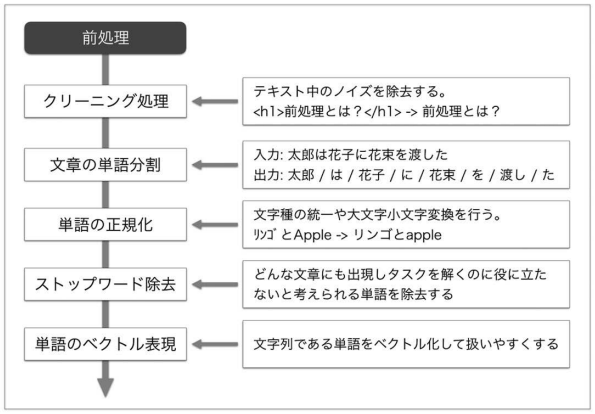
- 人間の発話をカテゴリごとに分類するモデルを決定木で作成します。

演習1：独自データ収集

- 下記の問いかけの発話事例を記載してください。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：BoWの実装

- 文章をベクトル化するプログラムを実装してください。

演習3：決定木による分類モデル

- ベクトル化したデータで決定木モデルを作成し、モデルの精度を確認してください。

決定木による分類結果

- 下記の表は決定木によるモデルの分類結果の一例です。
- 「Class_2：好きな食べ物を聞かれる」の誤認識が多いことがわかります。
- 次回の講義では、モデル精度を向上させるための工夫について学んでいきます。

テストデータに対する正解率： 0.88

predict	0	1	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	0	0	1
3	0	0	3	0
4	0	0	0	2

コーパスの充実

- 学習データに表記揺れが含まれた方が、頑健なモデルが作成できる場合があります（※逆の発想で、ベクトル化の際にこのような表記揺れを落としてしまう手法もあります）。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ ている ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

演習4：独自データの拡充

- データ数を増やして、モデルの精度の変化を確認してください。

```
sentences = [
    "名前は何",
    "名前は何ですか",
    "名前を何ていうの",
    "何ていうの名前は",
    "何ですか名前は",
    "名前を何ていうの",
    "あなたのお名前は",
    "お名前はあなたの",
    "お名前教えて",
    "お名前教えてよ",
```

表現の揺れを足してみる

決定木による分類結果

- 決定木でモデルを作成し、分類結果を考察しました。
- 全体の精度は向上し「Class_2：好きな食べ物を聞かれる」の誤認識も改善しました。

テストデータに対する正解率: 0.88

predict	0	1	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	0	0	1
3	0	0	3	0
4	0	0	0	2

テストデータに対する正解率: 0.93

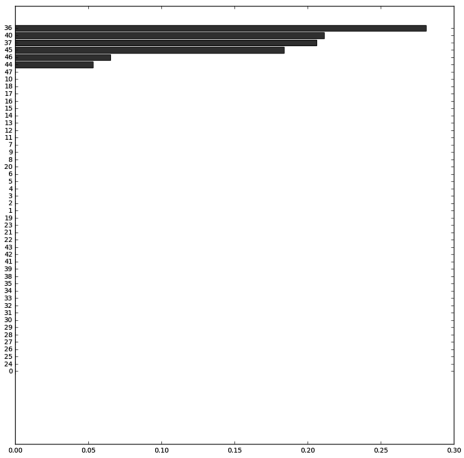
predict	0	1	2	3	4
class					
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	3	0	1
3	0	0	0	5	0
4	0	0	0	0	1

演習5：項目の寄与度

- 決定木モデルの判定に寄与した項目を確認してください。

判定に寄与した項目の確認

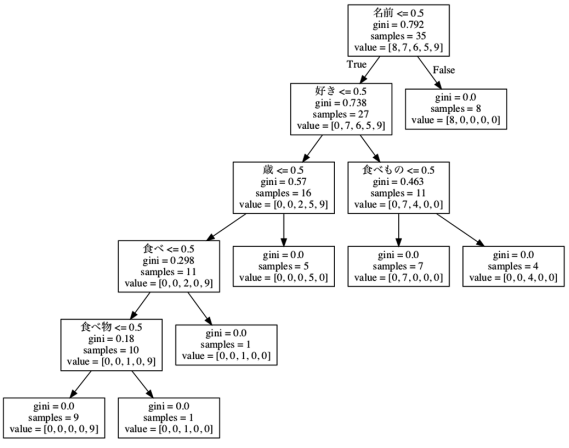
- 決定木における判定に寄与した項目を数値化した結果です。赤字の項目のみが判定に寄与し、その他の項目は判定に寄与していないことが確認できます。



{'あなた': 0, 'いくつ': 1, 'お': 2, 'おはよう': 3, 'おやすみ': 4, 'おやすみなさい': 5, 'か': 6, 'が': 7, 'こんち': 8, 'こんにちは': 9, 'こんばんは': 10, 'ござい': 11, 'さようなら': 12, 'す': 13, 'た': 14, 'だい': 15, 'ちょうだい': 16, 'て': 17, 'ていう': 18, 'です': 19, 'と': 20, 'どんな': 21, 'な': 22, 'なっ': 23, 'なに': 24, 'に': 25, 'の': 26, 'は': 27, 'ます': 28, 'よ': 29, 'れる': 30, 'を': 31, 'セロリ': 32, 'ピーマン': 33, 'ー': 34, '何': 35, '名前': 36, '好き': 37, '思う': 38, '教え': 39, '歳': 40, '美味しい': 41, '色': 42, '赤色': 43, '食べ': 44, '食べもの': 45, '食べ物': 46, '黄色': 47}

判定に寄与した項目の確認

- 寄与度の大きな項目で分岐していることが確認できます。



線形回帰

アジェンダ

- 振り返り
- 線形回帰モデルの構築

振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

線形回帰モデルの構築

線形回帰とは

- 線形回帰は連続値をとる目的変数yと説明変数x（特徴量）の関係を下記の数式でモデル化します（ $X_0=1$とし、w_0は切片を表します。）。
- 説明変数が一つの場合を単回帰、複数の場合を重回帰といいます。

$$y = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i$$

演習：線形回帰による住宅価格の予測モデルの構築

- ボストン市郊外の地域別住宅価格を予測する線形回帰モデルを作成します。

演習1：データ項目の確認

- ボストン市郊外の地域別住宅価格データの項目を確認してください。

変数	説明
CRIM	町ごとの一人当たりの犯罪率
ZN	25,000平方フィートを超える敷地に区画された宅地の割合
INDUS	非小売業種の土地面積の割合
CHAS	Charles Riverダミー変数（敷地が川の境界にある場合は1、それ以外の場合は0）
NOX	窒素酸化物の濃度（1000万分の1）
RM	1住戸あたりの平均部屋数
AGE	1940年以前に建設された所有者居住ユニットの割合
DIS	ボストンの5つの雇用センターまでの重み付き距離
RAD	ラジアルハイウェイ（放射状に各方面へ伸びる高速道路）へのアクセスのしやすさの指標
TAX	10,10,000ドルあたりの全額固定資産税率
PTRATIO	町による生徒 - 教師比率
B	1000（Bk - 0.63）^ 2ここでBkは町による黒人の割合
LSTAT	低所得者の割合
MEDV	住宅価格の中央値（1,000単位）

演習2：項目間の相関

- MEDV（住宅価格の中央値）と他の項目の相関を確認してください。

項目間の相関

- MEDV（住宅価格の中央値）とRM（1住戸あたりの平均部屋数）は比較的強い正の相関があることがわかります。
- MEDV（住宅価格の中央値）とLSTAT（低所得者の割合）は比較的強い負の相関があることがわかります。

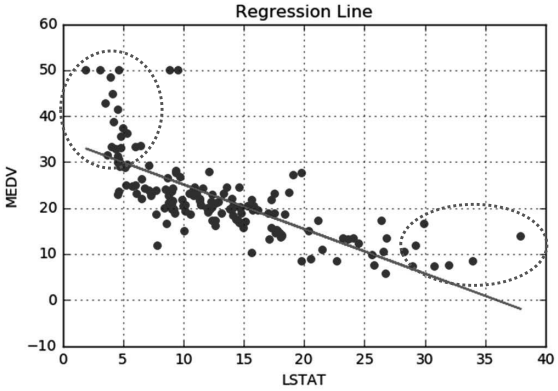
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

演習3：単回帰モデル

- 住宅価格の中央値を低所得者の割合から予測する線形回帰モデルを構築し、精度を確認してください。

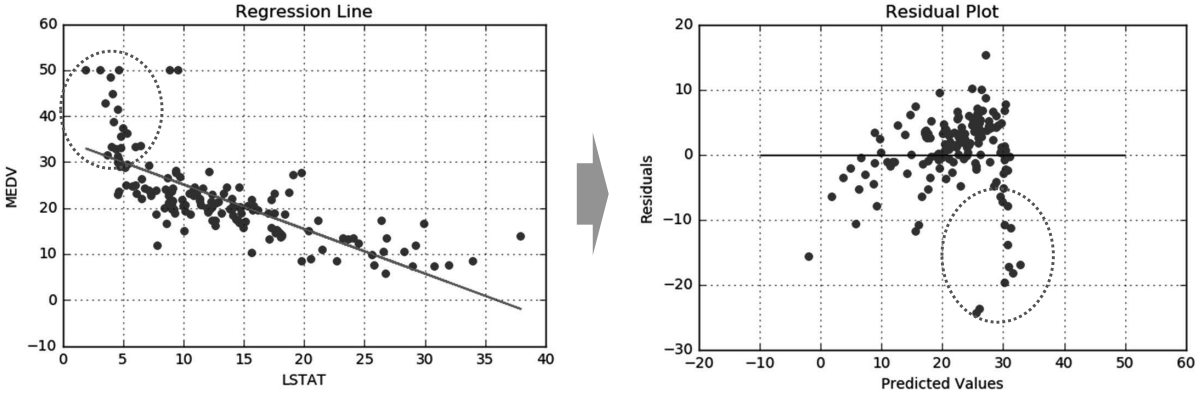
回帰線の確認

- 目的変数である「MEDV: 住宅価格」と比較的強い負の相関がある説明変数である「LSTAT: 低所得者の割合」の散布図と、予測線を図示します。
- LSTATが7～28の範囲では、予測線は住宅価格をよく表現できていますが、その範囲外では乖離があることが確認できます。



残差の確認

- 目的変数である「MEDV: 住宅価格」と予測線の差分（残差）を図示してみます。
- 残差が0を中心にばらついていれば良いモデルが作れたと言えます。今回はLSTATが7～28の範囲外にあるデータへの残差が大きなモデルとなっていることが確認できます。



モデル性能の評価

- モデルの性能を評価するために、何かしらの指標を設定したほうが便利です。線形回帰モデルの性能評価として、下記の指標を用いることが一般的です。
  - 平均二乗誤差：残差平方和をデータ数で正規化した値
  - 決定係数：相関係数の二乗

```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

```
r^2 train data:  0.5524780757890007
r^2 test data:  0.5218049526125568
```

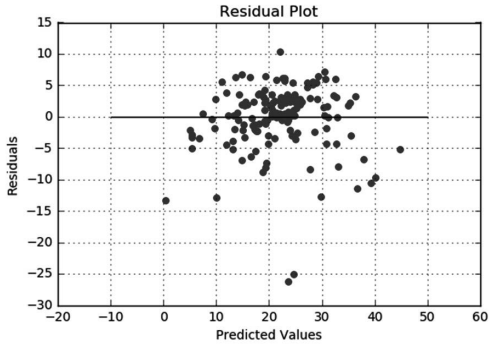
決定係数の例。モデル作成に使用した学習データに対する決定係数が、テストデータに対する決定係数より若干高いことが確認できます。

演習4：重回帰モデル

- 住宅価格の中央値を全項目から予測する線形回帰モデルを構築し、精度を確認してください。

重回帰モデルの精度

- 説明変数を追加したことにより、決定係数が大幅に改善されていることが確認できます。
- モデルの精度向上のみが目的であれば説明変数を増やして精度向上を図るのは1つの方法ですが、「過学習」の問題が発生することがあります。
- また、モデルの出力結果への説明変数の寄与度を正しく評価できなくなる「多重共線性」の問題が発生することがあります。




```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

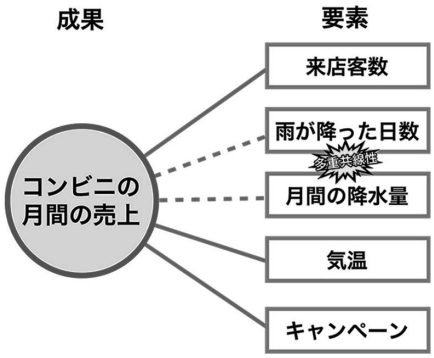
```
r^2 train data:  0.7644563391821222
r^2 test data:  0.6735280865347231
```

多重共線性

- 説明変数を増やしていくと一般的にモデルの表現力が向上し、精度が向上します。
- モデルの精度を高めることのみが目的であれば支障がないこともありますが、モデルの説明性（モデルはなぜそのような予測をしたのか、の説明）が問われる場合、説明変数を闇雲に増やすことには注意が必要です。

多重共線性

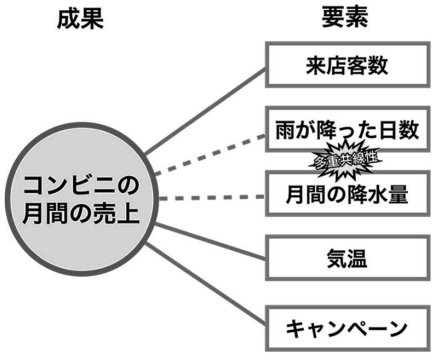
- 説明変数間で相関係数が高い時に多重共線性（multicollinearity）という問題が発生します。
- 多重共線性とは、モデル式の係数が不安定（符号と大きさが安定しない）になり、モデルの予測結果に対する係数の寄与度を正しく評価することができなくなってしまいます。



出展：https://xica.net/vno4ul5p/

多重共線性

- 多重共線性の回避策としては、相関が高い係数のどちらか一方をモデルから外す、ことが一般的です。



出展：https://xica.net/vno4ul5p/

多重共線性の事例

- 説明変数に全項目を使用した重回帰モデルの係数を表1に示します。INDUSとNOXの符号が逆になっているのが確認できます。

表1：重回帰モデルの係数

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.119859	0.0444233	0.0118612	2.51295	-16.271	3.8491	-0.00985472	-1.50003	0.241508	-0.0110672	-1.01898	0.00695273	-0.488111

符号が逆になっている。

表2：全項目の相関係数

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007366	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

INDUSとNOXは正の相関がある。

MEDV（住宅価格）に対し、INDUSとNOXは負の相関がある。

過学習、多重共線性の回避

- 過学習や多重共線性を回避するために正則化という手法が存在します。
  - L1正則化：いくつかの説明変数の係数を0にする手法（特徴選択を行っていることになる）です。スパース（疎な）な行列で表現するため、高速に計算できるようになる。
  - L2正則化：各説明変数の係数が大きくなりすぎないようにする（個々の特徴量が出力に与える影響をなるべく小さくした）手法です。

演習5：L1正則化

- モデルにL1正則化を適用し、精度を確認してください。

演習6：L2正則化

- モデルにL2正則化を適用し、精度を確認してください。

正則化の効果

- L1正則化を実施することでINDUS、CHAS、NOXの係数が0となっていることが確認できます。
- L2正則化を実施することで係数の大きさが均されているのが確認できます。またINDUSとNOXの符号がちぐはぐになっていた問題が解消されていることが確認できます。
- 今回の例では、正則化を実施した影響で精度が若干低下していることも確認できます。精度向上を目的とするのか、モデルの説明性を高めることを目的とするのかで正則化を実施するか否かが異なってきます。

正則化なし

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.119859	0.0444233	0.0118612	2.51295	-16.271	3.8491	-0.00985472	-1.50003	0.241508	-0.0110672	-1.01898	0.00695273	-0.488111

r^2 train data: 0.7644563391821222
r^2 test data: 0.6735280865347231

L1正則化後

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.0660405	0.048338	-0	0	-0	0.864003	0.0122097	-0.751367	0.200142	-0.0139477	-0.846463	0.00676481	-0.733092

r^2 train data: 0.7083629297638161
r^2 test data: 0.6114196752800867

L2正則化後

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.116678	0.0460807	-0.0204146	2.46073	-8.27864	3.88801	-0.017804	-1.39675	0.217637	-0.0116303	-0.932674	0.00740582	-0.495456

r^2 train data: 0.7622445226799686
r^2 test data: 0.6667168412456653

主成分分析

アジェンダ

- 振り返り
- 主成分分析アルゴリズムの実装

振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

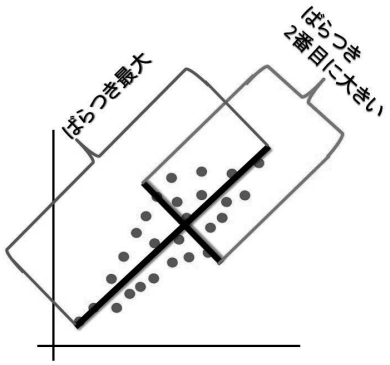
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

主成分分析アルゴリズムの実装

主成分分析とは

- 分散が最大になるように主成分軸を引くことを主成分分析といいます。
- 最も長い軸を第1主成分軸、次に長い軸を第2主成分軸といいます。
- 主成分分析を実施すると、主成分軸を通して多次元のデータを要約することができます。



出展： https://logics-of-blue.com/principal-components-analysis/

演習1：主成分分析の実施

- 2次元のダミーデータを作成し、主成分分析を実施してください。

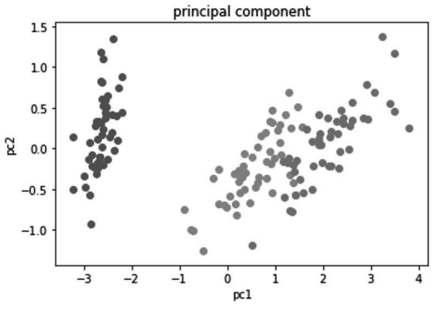
演習2：次元圧縮

- アイリスデータで主成分分析を実施してください。

次元の圧縮

- アイリスデータの特徴部分の次元は4次元です。主成分分析を実施して2次元に圧縮したのが右図です。
- 様々なデータについて分類器を作成する際は次元を圧縮しないほうが精度が高くなることが多いですが、次元を圧縮して単純化することによって、データの特性に対する気付きが得られることがあります。

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2]])
```



2019 年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

■実施委員会

◎	船山　世界	日本電子専門学校　校長
	大川　晃一	日本電子専門学校　エンジニア教育部長 ／ケータイ・アプリケーション科科長
	種田　裕一	東北電子専門学校　第２教務部長　学生サポート室長
	勝田　雅人	トライデントコンピュータ専門学校　校長
	安田　圭織	学校法人上田学園　上田安子服飾専門学校
	平田　眞一	学校法人第一平田学園　理事長
	平井　利明	静岡福祉大学　特任教授
	木田　徳彦	株式会社インフォテックサーブ　代表取締役
	渡辺　登	合同会社ワタナベ技研　代表社員
	岡山　保美	株式会社ユニバーサル・サポート・システムズ　取締役
	冨田　慎一郎	株式会社ウチダ人材開発センタ　常務取締役

■調査委員会

◎	大川　晃一	日本電子専門学校　エンジニア教育部長 ／ケータイ・アプリケーション科科長
	菊嶋　正和	株式会社サンライズ・クリエイティブ　代表取締役
	柴原　健次	合同会社ヘルシーブレイン　代表 CEO
	上田　あゆ美	株式会社ウチダ人材開発センタ

■人材育成委員会

◎	大川　晃一	日本電子専門学校　エンジニア教育部長 ／ケータイ・アプリケーション科科長
	福田　竜郎	日本電子専門学校　AI システム科
	阿保　隆徳	東北電子専門学校　学科主任
	小澤　慎太郎	中央情報大学校　高度情報システム学科
	神谷　裕之	名古屋工学院専門学校　メディア学部　情報学科
	北原　聡	麻生情報ビジネス専門学校　校長代行
	原田　賢一	有限会社ワイズマン　代表取締役
	柴原　健次	合同会社ヘルシーブレイン　代表 CEO
	菊嶋　正和	株式会社サンライズ・クリエイティブ　代表取締役

2019 年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

教員用研修プログラム

令和２年２月

学校法人電子学園（日本電子専門学校）
〒169-8522　東京都新宿区百人町 1-25-4
TEL 03-3369-9333　FAX 03-3363-7685