2019年度「専修学校による地域産業中核的人材養成事業」

AIプログラミングⅡ教材

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」

AI プログラミング II 教材

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

目 次

第1回：データ前処理

前処理の必要性と種類

アジェンダ

- データの種類
- データ前処理とは
- 機械学習の手法の分類
  - 数値データ
  - 区分値
  - 自然言語
  - 画像

全15回の講義について

- 主にデータ前処理の種類、実行方法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

データの種類

データの種類：数値データ

- アヤメデータを例に取ります。アヤメデータは花弁とがく片の長さと幅を、アヤメの種類ごとに記録したデータです。
- 左から4列（sepal_length/sepal_width/petal_length/petal_width）は数値が記録されています。このようにデータを数値データといいます。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。

アヤメの種類。

データの種類：区分値

- 一番右の列（species）は言語でアヤメの種類が記録されています。このようにデータを区分値といいます。
- 下の例では言語で記録されていますが、仮にsetosaは1、versicolorは2、virginicaは3として1～3の数字で記録されていた場合でも、数値データとは意味合いが違いますので、区分値として扱います。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。

アヤメの種類。

データの種類：自然言語

- 自然言語（人間が話す言葉）は長さ、意味が多岐に渡ります。数値しか認識できない機械学習器で自然言語処理をそのまま扱うことはできません。

自然言語の形態素解析例

```
今日の横浜の天気を教えて
今日	名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
の	助詞,連体化,*,*,*,*,の,ノ,ノ
横浜	名詞,固有名詞,地域,一般,*,*,横浜,ヨコハマ,ヨコハマ
の	助詞,連体化,*,*,*,*,の,ノ,ノ
天気	名詞,一般,*,*,*,*,天気,テンキ,テンキ
を	助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
教え	動詞,自立,*,*,一段,連用形,教える,オシエ,オシエ
て	助詞,接続助詞,*,*,*,*,て,テ,テ
EOS
```

データの種類：画像

- 画像はサイズ（縦横サイズ）、画素（画像の粗さの指標）、色情報（R：赤、G：緑、B：青、A：透明度）で構成されています。
- 数値しか認識できない機械学習器で画像をそのまま扱うことはできません。

画像処理例でよく用いられる女性の写真

データの種類：その他のデータ

- 音声データ。音の強弱が振幅として現れた時系列波形データとして記録されます。
- 動画データ。画像が時間軸で連続して入力され、また音声データが加わったデータです。

音声データの例
出展：https://tools4music.info/audio-editer/page/2

データの前処理

データの前処理とは

- 「学習」ステップにおいて機械学習などを活用してモデルを作成しますが、その前ステップとして「前処理」が必要となります。
- 前処理は、機械学習器を使用するためには必須である前処理と、モデル精度を向上させるために実施することが望ましい前処理とに大別できます。

モデル作成のステップ

必須の前処理とは

- 機械学習器は数値を扱うことができます。区分値や自然言語、画像などは機械学習器で扱える数値データに変換する必要があります。
- 例え数値データを扱うとしても、欠損値が存在する場合はそのままでは機械学習器に掛ける事ができない場合があります。

必須の前処理とは

- モデルの作成と評価を行うためには、データセットを4分割する必要があります。
- 学習データでモデルを作成し、テストデータでモデルの精度を評価します。

※本資料では、広義の意味で学習/テストデータセット作成も「データ前処理」に含めます。

モデル精度向上のための前処理とは

- モデル精度向上のための前処理は多岐に渡りますが、一例を紹介します。
- アヤメデータにおいてpetal_widthは1未満の数値ですが、sepal_lengthは4より大きな数値となっています。このようなデータセットに対し、全ての列で最大値が1になるような正規化を実施すると、モデルの精度が向上することがあります。

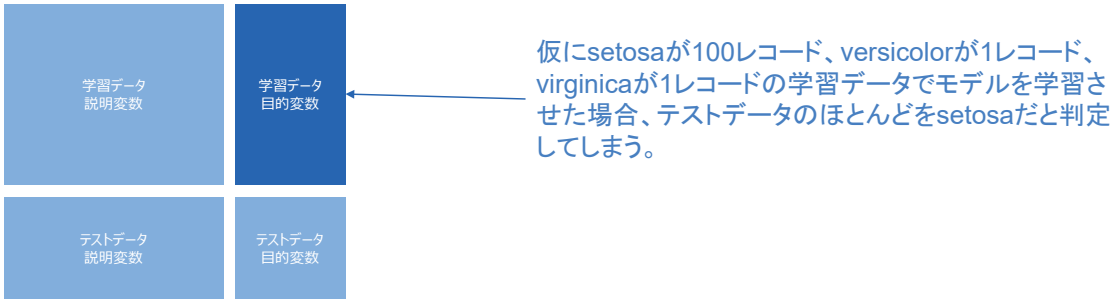
アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

花弁とがく片の長さ、幅のデータ。

モデル精度向上のための前処理とは

- 学習データの目的変数の分布が大きく偏っている場合に分類器を作成すると、分布が大きなクラスに偏ったモデルが作成されてしまいます。そのような場合、目的変数のクラスの分布を整えるとモデル精度が向上する場合があります。



第2回：統計量1

前処理時に確認する統計量

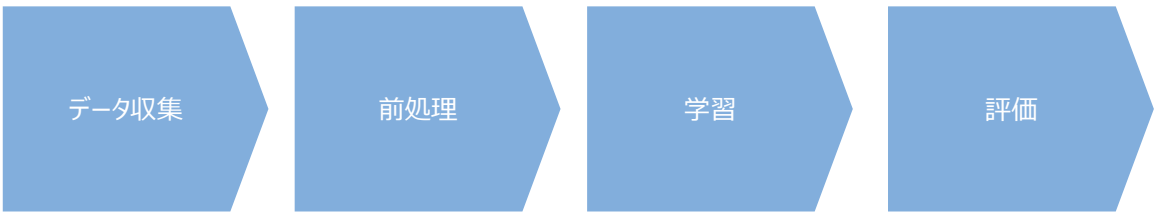
アジェンダ

- 第1回の振り返り
- データの統計量
  - データ数（標本数）
  - 平均値
  - 最小値
  - 最大値
  - 中央値
  - 最頻値

第1回の振り返り
データの前処理の概要

データの前処理とは

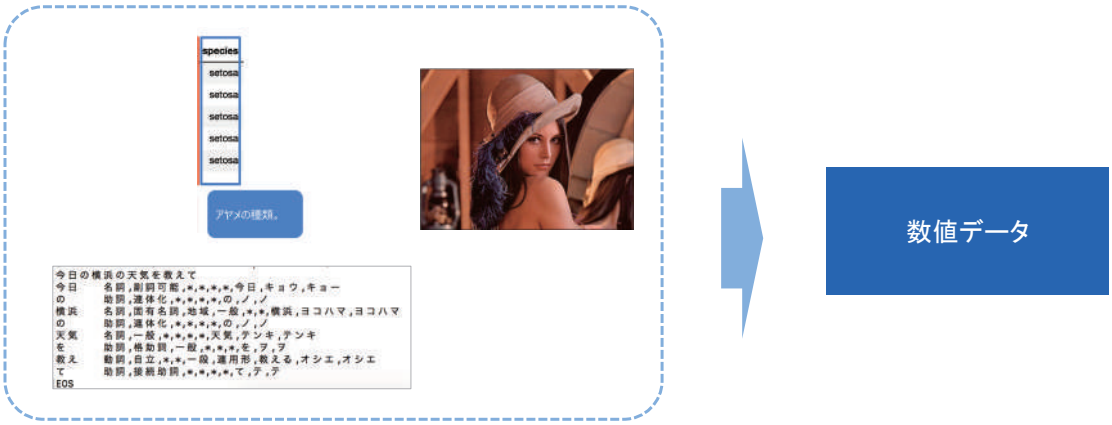
- 「学習」ステップにおいて機械学習などを活用してモデルを作成しますが、その前ステップとして「前処理」が必要となります。
- 前処理は、機械学習器を使用するためには必須である前処理と、モデル精度を向上させるために実施することが望ましい前処理とに大別できます。



モデル作成のステップ

必須の前処理とは

- 機械学習器は数値を扱うことができます。区分値や自然言語、画像などは機械学習器で扱える数値データに変換する必要があります。
- 例え数値データを扱うとしても、欠損値が存在する場合はそのままでは機械学習器に掛ける事ができない場合があります。



必須の前処理とは

- モデルの作成と評価を行うためには、データセットを4分割する必要があります。
- 学習データでモデルを作成し、テストデータでモデルの精度を評価します。

※本資料では、広義の意味で学習/テストデータセット作成も「データ前処理」に含めます。

モデル精度向上のための前処理とは

- モデル精度向上のための前処理は多岐に渡りますが、一例を紹介します。
- アヤメデータにおいてpetal_widthは1未満の数値ですが、sepal_lengthは4より大きな数値となっています。このようなデータセットに対し、全ての列で最大値が1になるような正規化を実施すると、モデルの精度が向上することがあります。

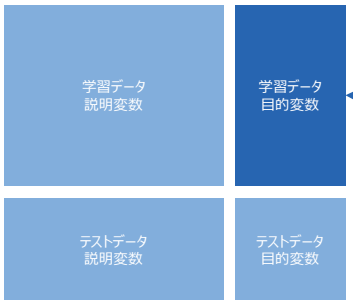
アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

花弁とがく片の長さ、幅のデータ。

モデル精度向上のための前処理とは

- 学習データの目的変数の分布が大きく偏っている場合に分類器を作成すると、分布が大きなクラスに偏ったモデルが作成されてしまいます。そのような場合、目的変数のクラスの分布を整えるとモデル精度が向上する場合があります。



仮にsetosaが100レコード、versicolorが1レコード、virginicaが1レコードの学習データでモデルを学習させた場合、テストデータのほとんどをsetosaだと判定してしまう。

データの統計量

データ数（標本数）

- モデルを作成するためのデータを入手したら、まずはデータ数を確認します。10クラスの分類をしたいのにデータ数が10しかなければ、モデルを作ることはできません（作っても全く当てにならないモデルになります）。
- 後段の講義（不均衡データへの対応）で取り上げますが、2クラス分類したい場合に手元のデータが1クラス分しか無い場合も、モデルを作ることが困難になります。

- データを手に入れたら、まずはデータ数を確認します。データが数値データであればレコード数を確認し、画像であれば画像ファイルの数を確認します。

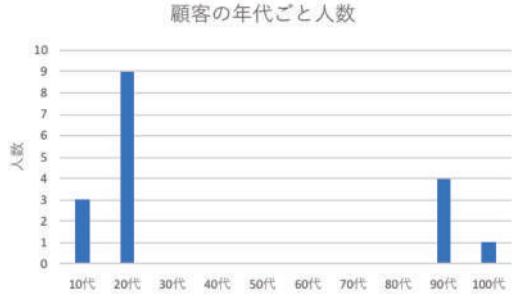
演習1：データ数の確認

- アヤメデータのレコード数を確認してください。

平均値

- 算術平均や相加平均とも呼ばれます。データの合計÷データの数で求めます。

- 平均は統計量として頻繁に用いられますが、対象データの特徴をよく表す統計量とはならないこともあるので注意が必要です。
- 例えば、ある担当者の顧客の年齢データを確認すると、18歳×3名、21歳×6名、23歳×3名、95歳×2名、98歳×2名、102歳×1名だったとします。平均は43歳ですが、その年代の顧客は全く存在しないので、40歳代向けの販促活動をしても効果は望めません。



演習2：平均値の確認

- アヤメデータの数値列について平均値を確認してください。

最小値

- データの中の最も小さい値のことです。
- 最小値を確認することによって、「その数値列はマイナスの値を持つのか」ということを確認することができます。

演習3：最小値の確認

- アヤメデータの数値列について最小値を確認してください。

最大値

- データの中の最も大きい値のことです。
- 最小値とともに最大値を確認することによって、「その数値列はどの範囲の値を持つのか」ということを確認することができます。

演習4：最大値の確認

- アヤメデータの数値列について最大値を確認してください。

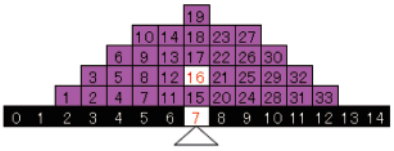
中央値

- データを昇順で並び替えた際、真ん中にくる値のことです。データの個数が奇数の場合はちょうど真ん中の数値であり、データの個数が偶数の場合は真ん中の2つの数値の平均値とします。
- 例えば、データの集合{1,2,3,4,5}ならば中央値3、データの集合{1,2,3,4}ならば中央値2.5となります。

中央値と平均値の違い

- 「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。
- 最小値が2、最大値が12の33個のデータがあったとします。33個のデータを小さい方から順に1番から33番まで番号を振っておきます。一本の定規の上に、各データを、データが持っている値と定規の目盛りが一致するように積み上げたところをイメージしてください。

- この定規（上図の黒い棒）の左右のバランスが取れるところ、この例では７が平均値です。平均値とはてこの原理における支点です。一方、中央値はデータの並びにおいてちょうど真ん中のところ、この例なら16番のデータの置かれた7が中央値になります。このように平均値を中心に左右均等に散らばる場合は平均値も中央値も同じになります。



中央値と平均値の違い

- （前ページに引き続き「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。）

- それでは33番のデータの値が12ではなく45だったとしましょう。33番が右に大きくずれたことで、てこの原理が働いて平均値は8になります。平均値は外れ値の影響を受けやすいことが分かります。中央値は外れ値の影響を受けないので7のままです。

- 33番のデータの値が45ではなく450だったとしても中央値は同じですね。でも平均値は20を超えてしまいます。こうなると、33個中32個は平均値より下ということになってしまいます。 一方、中央値は、依然、7のままですね。中央値のこの性質のことを「外れ値に対してロバストである（頑健性がある）」と言います。

演習5：中央値の確認

- アヤメデータの数値列について中央値を確認してください。

最頻値

- 最頻値とは、データの集合の中で、最も頻繁に現れる値のことです。例えば、データの集合{1,2,2,3,4,5}ならば最頻値は2となります。

- ただし、一般に数値データの集合の場合、同じ数値が頻繁に現れることは稀です。そのような場合は数値データの度数分布表を作成し、最も度数が高い階級の階級値を最頻値とします。※度数分布表の階級の取り方により、最頻値が変わってしまうことに注意してください。

演習6：最頻値の確認

- アヤメデータの数値列について最頻値を確認してください。

演習7：ヒストグラムの確認

- アヤメデータの数値列についてヒストグラムを作成、データの散らばりを確認してください。

第3回：統計量2

前処理時に確認する統計量

アジェンダ

- 第2回の振り返り
- データの統計量
  - 歪度
  - 尖度
  - 分散
  - 標準偏差
  - 変動係数

第2回の振り返り
統計量1

データ数（標本数）

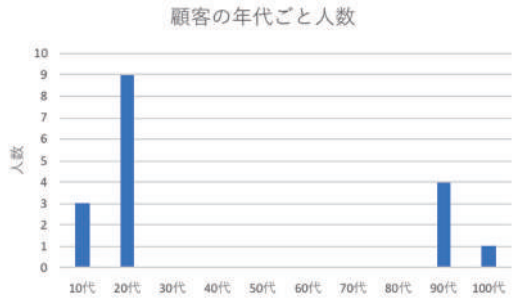
- モデルを作成するためのデータを入手したら、まずはデータ数を確認します。10クラスの分類をしたいのにデータ数が10しかなければ、モデルを作ることはできません（作っても全く当てにならないモデルになります）。
- 後段の講義（不均衡データへの対応）で取り上げますが、2クラス分類したい場合に手元のデータが1クラス分しか無い場合も、モデルを作ることが困難になります。

- データを手に入れたら、まずはデータ数を確認します。データが数値データであればレコード数を確認し、画像であれば画像ファイルの数を確認します。

平均値

- 算術平均や相加平均とも呼ばれます。データの合計÷データの数で求めます。

- 平均は統計量として頻繁に用いられますが、対象データの特徴をよく表す統計量とはならないこともあるので注意が必要です。
- 例えば、ある担当者の顧客の年齢データを確認すると、18歳×3名、21歳×6名、23歳×3名、95歳×2名、98歳×2名、102歳×1名だったとします。平均は43歳ですが、その年代の顧客は全く存在しないので、40歳代向けの販促活動をしても効果は望めません。



最小値

- データの中の最も小さい値のことです。
- 最小値を確認することによって、「その数値列はマイナスの値を持つのか」ということを確認することができます。

最大値

- データの中の最も大きい値のことです。
- 最小値とともに最大値を確認することによって、「その数値列はどの範囲の値を持つのか」ということを確認することができます。

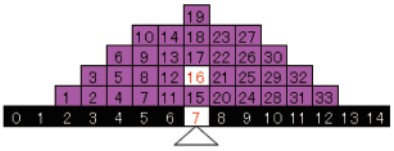
中央値

- データを昇順で並び替えた際、真ん中にくる値のことです。データの個数が奇数の場合はちょうど真ん中の数値であり、データの個数が偶数の場合は真ん中の2つの数値の平均値とします。
- 例えば、データの集合{1,2,3,4,5}ならば中央値3、データの集合{1,2,3,4}ならば中央値2.5となります。

中央値と平均値の違い

- 「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。
- 最小値が2、最大値が12の33個のデータがあったとします。33個のデータを小さい方から順に1番から33番まで番号を振っておきます。一本の定規の上に、各データを、データが持っている値と定規の目盛りが一致するように積み上げたところをイメージしてください。

- この定規（上図の黒い棒）の左右のバランスが取れるところ、この例では７が平均値です。平均値とはてこの原理における支点です。一方、中央値はデータの並びにおいてちょうど真ん中のところ、この例なら16番のデータの置かれた7が中央値になります。このように平均値を中心に左右均等に散らばる場合は平均値も中央値も同じになります。



中央値と平均値の違い

- （前ページに引き続き「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。）

- それでは33番のデータの値が12ではなく45だったとしましょう。33番が右に大きくずれたことで、てこの原理が働いて平均値は8になります。平均値は外れ値の影響を受けやすいことが分かります。中央値は外れ値の影響を受けないので7のままです。

- 33番のデータの値が45ではなく450だったとしても中央値は同じですね。でも平均値は20を超えてしまいます。こうなると、33個中32個は平均値より下ということになってしまいます。 一方、中央値は、依然、7のままですね。中央値のこの性質のことを「外れ値に対してロバストである（頑健性がある）」と言います。

最頻値

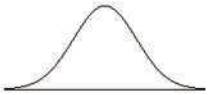
- 最頻値とは、データの集合の中で、最も頻繁に現れる値のことです。例えば、データの集合{1,2,2,3,4,5}ならば最頻値は2となります。
- ただし、一般に数値データの集合の場合、同じ数値が頻繁に現れることは稀です。そのような場合は数値データの度数分布表を作成し、最も度数が高い階級の階級値を最頻値とします。※度数分布表の階級の取り方により、最頻値が変わってしまうことに注意してください。

データの統計量2

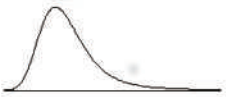
歪度

- 分布の非対称性を表す値のことです。

・歪度＜0
左に裾が長い（右に偏った）分布



・歪度＝0
正規分布のとき歪度は0



・歪度＞0
右に裾が長い（左に偏った分布）

出展：https://www.trifields.jp/statistical-analysis-basic-statistics-164

演習1：歪度の確認

- アヤメデータの数値列について歪度を確認してください。
- ヒストグラムと歪度を比較し、数値の大小と分布の偏りを確認してください。

尖度

- 分布の尖り度合いを表す値のことです。
  - 尖度＜0：なだらかな分布
  - 尖度＝0：正規分布のときは尖度は3
  - 尖度＞0：尖っている分布

演習2：尖度の確認

- アヤメデータの数値列について尖度を確認してください。
- ヒストグラムと尖度を比較し、数値の大小と分布の偏りを確認してください。

分散

- 平均値からの散らばり具合を表す数値を分散といい、次式で定義されます。

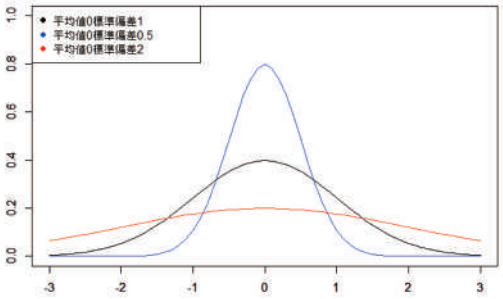
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

演習3：分散の確認

- アヤメデータの数値列について分散を確認してください。

標準偏差

- 分散の正の平方根値のことです。



出展：https://www.trifields.jp/statistical-analysis-basic-statistics-164

演習4：標準偏差の確認

- アヤメデータの数値列について標準偏差を確認してください。

変動係数

- 標準偏差を平均値で割った値のことで、単位の異なるデータのばらつきや、平均値に対するデータとばらつきの関係を相対的に評価する際に用いる単位を持たない（＝無次元の）数値です。変動係数はCVで表されることがあります。
- 下記の例では、一見すると牛ステーキ肉のばらつきが大きいように見えますが、変動係数で評価すると鶏ささみの方がばらつきが大きいことがわかります。

肉の種類	平均価格（円）	標準偏差（円）
鶏ささみ	80	20
牛ステーキ肉	1800	300

- 鶏ささみの変動係数：20÷80=0.25
- 牛ステーキ肉の変動係数：300÷1800≒0.167

出展：https://bellcurve.jp/statistics/course/5929.html

演習5：変動係数の確認

- アヤメデータの数値列について変動係数を確認してください。

第4回：連続値や離散値の加工

アジェンダ

- 第2回、第3回（データの統計量）の振り返り
- 連続値と離散値のデータ加工
  - Min-Max normalization
  - 標準化
  - 連続値の離散化
  - 区分値のダミー変数化

データ統計量の振り返り

データ数（標本数）

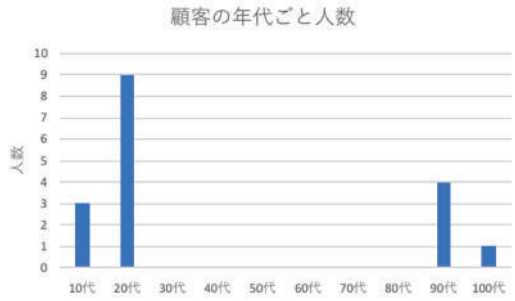
- モデルを作成するためのデータを入手したら、まずはデータ数を確認します。10クラスの分類をしたいのにデータ数が10しかなければ、モデルを作ることはできません（作っても全く当てにならないモデルになります）。
- 後段の講義（不均衡データへの対応）で取り上げますが、2クラス分類したい場合に手元のデータが1クラス分しか無い場合も、モデルを作ることが困難になります。

- データを手に入れたら、まずはデータ数を確認します。データが数値データであればレコード数を確認し、画像であれば画像ファイルの数を確認します。

平均値

- 算術平均や相加平均とも呼ばれます。データの合計÷データの数で求めます。

- 平均は統計量として頻繁に用いられますが、対象データの特徴をよく表す統計量とはならないこともあるので注意が必要です。
- 例えば、ある担当者の顧客の年齢データを確認すると、18歳×3名、21歳×6名、23歳×3名、95歳×2名、98歳×2名、102歳×1名だったとします。平均は43歳ですが、その年代の顧客は全く存在しないので、40歳代向けの販促活動をしても効果は望めません。



最小値

- データの中の最も小さい値のことです。
- 最小値を確認することによって、「その数値列はマイナスの値を持つのか」ということを確認することができます。

最大値

- データの中の最も大きい値のことです。
- 最小値とともに最大値を確認することによって、「その数値列はどの範囲の値を持つのか」ということを確認することができます。

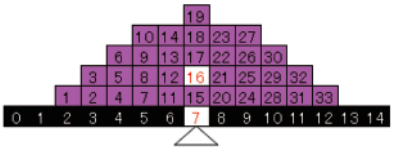
中央値

- データを昇順で並び替えた際、真ん中にくる値のことです。データの個数が奇数の場合はちょうど真ん中の数値であり、データの個数が偶数の場合は真ん中の2つの数値の平均値とします。
- 例えば、データの集合{1,2,3,4,5}ならば中央値3、データの集合{1,2,3,4}ならば中央値2.5となります。

中央値と平均値の違い

- 「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。
- 最小値が2、最大値が12の33個のデータがあったとします。33個のデータを小さい方から順に1番から33番まで番号を振っておきます。一本の定規の上に、各データを、データが持っている値と定規の目盛りが一致するように積み上げたところをイメージしてください。

- この定規（上図の黒い棒）の左右のバランスが取れるところ、この例では７が平均値です。平均値とはてこの原理における支点です。一方、中央値はデータの並びにおいてちょうど真ん中のところ、この例なら16番のデータの置かれた7が中央値になります。このように平均値を中心に左右均等に散らばる場合は平均値も中央値も同じになります。



中央値と平均値の違い

- （前ページに引き続き「https://bellcurve.jp/statistics/blog/14299.html」の記事を参考にします。）

- それでは33番のデータの値が12ではなく45だったとしましょう。33番が右に大きくずれたことで、てこの原理が働いて平均値は8になります。平均値は外れ値の影響を受けやすいことが分かります。中央値は外れ値の影響を受けないので7のままです。

- 33番のデータの値が45ではなく450だったとしても中央値は同じですね。でも平均値は20を超えてしまいます。こうなると、33個中32個は平均値より下ということになってしまいます。 一方、中央値は、依然、7のままですね。中央値のこの性質のことを「外れ値に対してロバストである（頑健性がある）」と言います。

最頻値

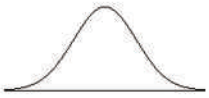
- 最頻値とは、データの集合の中で、最も頻繁に現れる値のことです。例えば、データの集合{1,2,2,3,4,5}ならば最頻値は2となります。

- ただし、一般に数値データの集合の場合、同じ数値が頻繁に現れることは稀です。そのような場合は数値データの度数分布表を作成し、最も度数が高い階級の階級値を最頻値とします。※度数分布表の階級の取り方により、最頻値が変わってしまうことに注意してください。

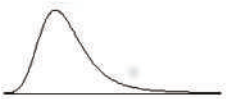
歪度

- 分布の非対称性を表す値のことです。

・歪度＜0
左に裾が長い（右に偏った）分布



・歪度＝0
正規分布のとき歪度は0



・歪度＞0
右に裾が長い（左に偏った分布）

出展：https://www.trifields.jp/statistical-analysis-basic-statistics-164

尖度

- 分布の尖り度合いを表す値のことです。
  - 尖度＜0：なだらかな分布
  - 尖度＝0：正規分布のときは尖度は3
  - 尖度＞0：尖っている分布

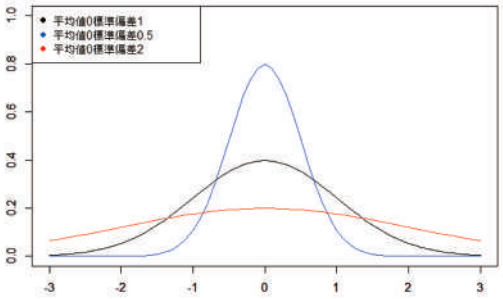
分散

- 平均値からの散らばり具合を表す数値を分散といい、次式で定義されます。

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

標準偏差

- 分散の正の平方根値のことです。



出展：https://www.trifields.jp/statistical-analysis-basic-statistics-164

変動係数

- 標準偏差を平均値で割った値のことで、単位の異なるデータのばらつきや、平均値に対するデータとばらつきの関係を相対的に評価する際に用いる単位を持たない（＝無次元の）数値です。変動係数はCVで表されることがあります。
- 下記の例では、一見すると牛ステーキ肉のばらつきが大きいように見えますが、変動係数で評価すると鶏ささみの方がばらつきが大きいことがわかります。

肉の種類	平均価格（円）	標準偏差（円）
鶏ささみ	80	20
牛ステーキ肉	1800	300

- 鶏ささみの変動係数：20÷80=0.25
- 牛ステーキ肉の変動係数：300÷1800≒0.167

出展：https://bellcurve.jp/statistics/course/5929.html

連続値と離散値のデータ加工

なぜ正規化（データ加工）が必要か

- モデル作成に説明変数（パラメータ）を複数使用する場合、説明変数間でそれぞれスケールが違うと、計算するときにスケールの違いに値が引っ張られてしまうことがあります。それを防ぐために正規化を実施します。

- 例えば、身長と体重が挙げられます。身長はcmで表現すると「178cm」の様に3桁の数値となり、体重はKgで表現すると「68Kg」のように2桁の数値となります。身長と体重を説明変数としたモデルを作成すると、数値の大きな身長により重みをおいたモデルになってしまいます。
- また、身長を「178cm」と数値そのままで表現するのではなく、「平均身長170cmからの差分」など何かしらの加工を施したほうがデータ群の特徴をよく表現できるようになることがあります。

Min-Max normalization

- 最小値を0、最大値を1とする正規化のことです。
- （Min-Max normalizationに限らずどの正規化にも共通していますが）データ内に外れ値が存在する場合、外れ値を処理した後に当該正規化を実施した方がモデルの精度は向上しやすくなります。

$$x_{new}^{i} = \frac{x^{i} - x_{min}}{x_{max} - x_{min}}$$

演習1：Min-Max normalization

- Min-Max normalizationを実装してください。

z-score normalization（標準化）

- 元データを平均0、標準偏差が1のデータに変換する正規化のことです。

$$x^i_{new} = \frac{x^i - \mu}{\sigma}$$

演習2：z-score normalization（標準化）

- z-score normalization（標準化）を実装してください。

連続値の離散化

- 連続値を何らかのカテゴリに分類する処理を離散化といいます。
- 例えば下記のように身長の範囲ごとに分類する処理のことをいいます。
  - グループ1：身長が～149cm
  - グループ2：身長が150～159cm
  - グループ3：身長が160～169cm
  - 、、、

演習3：連続値の離散化

- 連続値を離散化するプログラムを実装してください。

ダミー変数化

1次元の離散値列を、下記のように多次元のone hot vectorに変換する処理のことです。

setosa	versicolor	virginica
1	0	0
0	1	0
1	0	0
0	0	1

演習4：ダミー変数化

- 離散化データをダミー変数化するプログラムを実装してください。

第5回：データ補間と補外

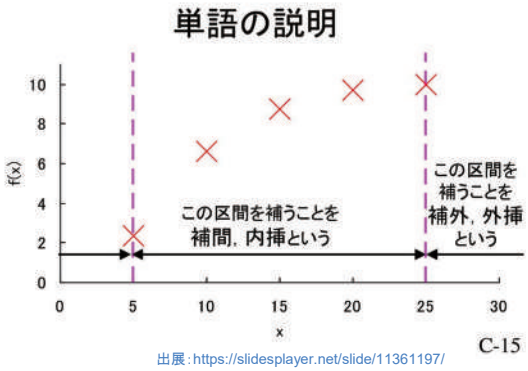
データ補強のアルゴリズム

アジェンダ

- 補間（内挿）と補外（外挿）
- 補間の種類
  - 線形補間
  - スプライン補間
  - 固定値補間
  - 最近傍補間

補間（内挿）と補外（外挿）

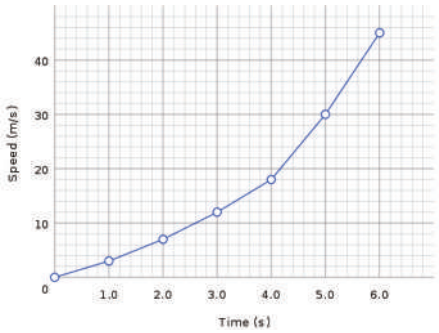
- 補間（内挿）とは、ある既知の数値データ列を基にして、そのデータ列の各区間の範囲内を埋める数値を求めること、またはそのような関数を与えることです。
- 補外（外挿）とは、ある既知の数値データを基にして、そのデータの範囲の外側で予想される数値を求めることです。



出展：https://slidesplayer.net/slide/11361197/

補間：線形補間

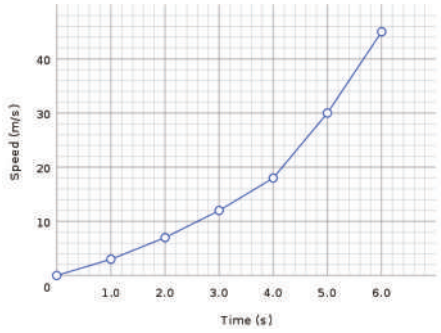
- 線形多項式（一次式）を用いた補間の手法です。



出展：https://ja.wikipedia.org/wiki/線形補間

演習1：線形補間

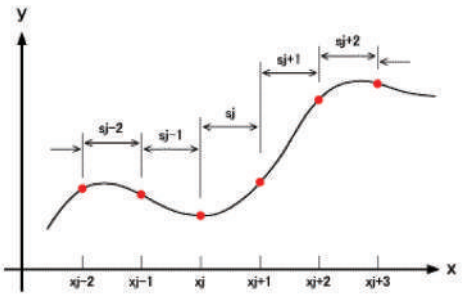
- 線形補間を実装してください。



出展：https://ja.wikipedia.org/wiki/線形補間

補間：スプライン補間

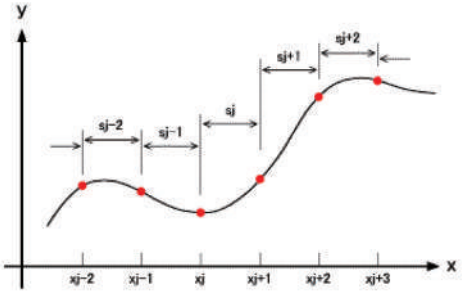
- スプラインを使用してデータを補間することです。スプラインとは区分多項式（区分的に定義された多項式）の事です。



出展：http://cannula.kir.jp/hokan.html

演習2：スプライン補間

- スプライン補間を実装してください。



出展：http://cannula.kir.jp/hokan.html

補間：固定値補間

- ある決まった値を全ての欠損値に代入する補間法です（数値列の欠損値全てに-1を代入するなど）。

演習3：固定値補間

- 固定値補間を実装してください。

補間：最近傍法

- 最も近いインデックスの値（最も近い前後の値）をそのまま欠損値への代入値としてコピーします。

演習4：最近傍法

- 最近傍法を実装してください。

第6回：機械学習データセット

モデル作成に必要なデータの構成

アジェンダ

- 説明変数と目的変数
- 学習データとテストデータ
- 見せかけの相関
- 多重共線性

データの加工：説明変数と目的変数の分離

- 通常、データセットは説明変数と目的変数が一体となっています。
- 機械学習モデルを作成する前に、どの列を目的変数として使用し、どの列を説明変数として使用するのかを決めます。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。
あやめの種類を判定するための説明変数として使用する。

あやめの種類。
目的変数として使用する。

演習1：説明変数と目的変数の分離

- アヤメデータを説明変数と目的変数に分離してください。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。
あやめの種類を判定するための説明変数として使用する。

あやめの種類。
目的変数として使用する。

データの加工：学習データとテストデータの分離

- モデルの作成と評価を行うためには、データセットを4分割する必要があります。
- 学習データでモデルを作成し、テストデータでモデルの精度を評価します。

演習2：学習データとテストデータの分離

- アヤメデータセットを学習データとテストデータに分離してください。

データの「量」と「質」の問題

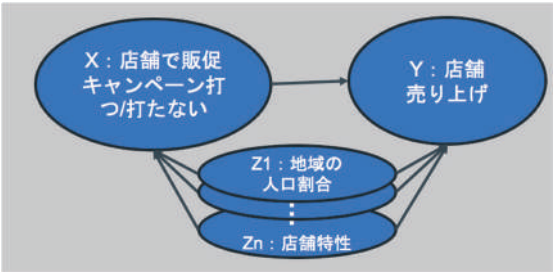
- 幾らパラメータを調整しても機械学習モデルの精度が思うように向上しない場合、データに問題があることが多々あります。
- データの問題とはデータが少ないという「量」の問題と、データに含まれる項目が適切でないという「質」の問題があります。

- データの量に問題がある場合は、データを蓄積することがモデル精度向上につながります。
- データの質に問題がある場合は、同じ形式のデータをそれ以上幾ら蓄積してもモデル精度向上は見込めません。

見せかけの相関

- データの質の問題を解決するには新たなデータ項目を導入する必要があります。

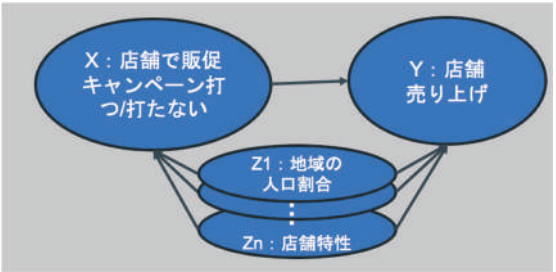
- 例えば下記のXを説明変数、Yを目的変数をするモデルがあるとします。日本全国でこのモデルは高い精度を出すのではなく、特定地域でしか高い精度を出さないモデルだとします。
- この例ではXとY両方に影響を及ぼすZ（交絡因子といいます）を明らかにし、モデルに取り込むことでモデルの精度向上が見込まれます。



出展：https://blog.datarobot.com/jp/causality_analysis_machine_learning2

見せかけの相関

- 交絡因子データの取得が困難な場合や、そもそも交絡因子がなにか定義が難しい場合は、モデルの適用範囲（X以外の条件がなるべく同じ条件）を明らかにした上でのモデル運用が好ましいです。



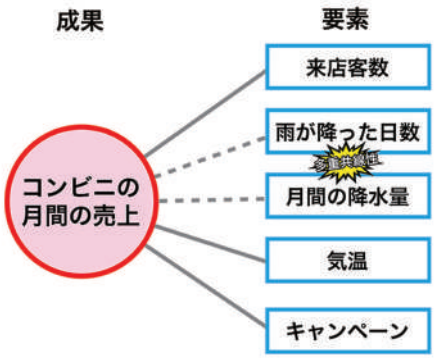
出展：https://blog.datarobot.com/jp/causality_analysis_machine_learning2

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

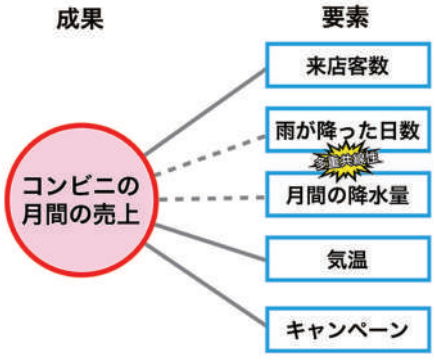
- 説明変数間で相関係数が高い時に多重共線性（multicollinearity）という問題が発生します。
- 多重共線性とは、モデル式の係数が不安定（符号と大きさが安定しない）になり、モデルの予測結果に対する係数の寄与度を正しく評価することができなくなってしまいます。



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/

第7回：不均衡データへの対応1

クラス数の確認

アジェンダ

- 第6回の振り返り
- 目的変数の分布

第6回の振り返り
機械学習データセット

データの加工：説明変数と目的変数の分離

- 通常、データセットは説明変数と目的変数が一体となっています。
- 機械学習モデルを作成する前に、どの列を目的変数として使用し、どの列を説明変数として使用するのかを決めます。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。
あやめの種類を判定するための説明変数として使用する。

あやめの種類。
目的変数として使用する。

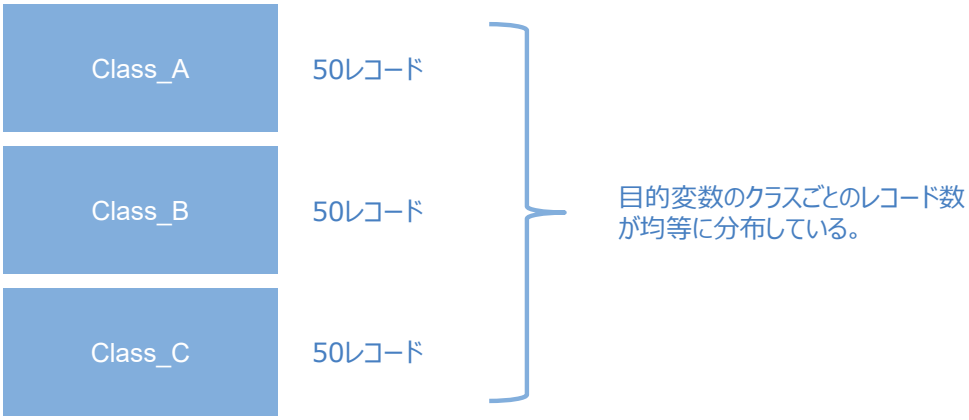
データの加工：学習データとテストデータの分離

- モデルの作成と評価を行うためには、データセットを4分割する必要があります。
- 学習データでモデルを作成し、テストデータでモデルの精度を評価します。

目的変数の分布

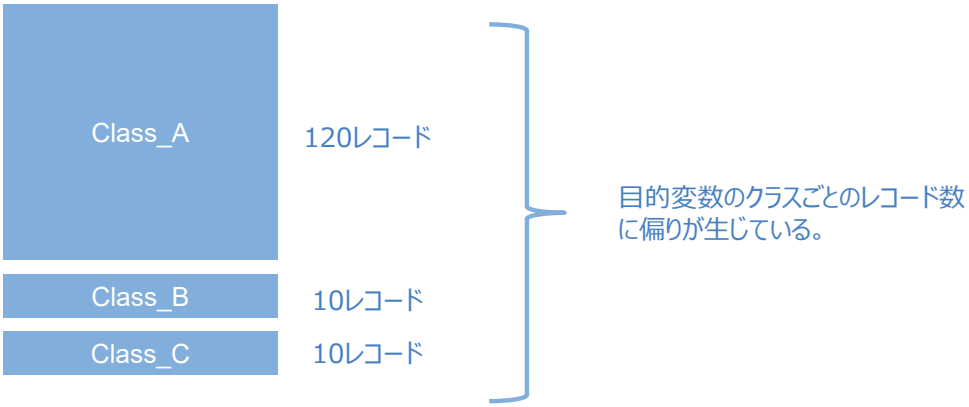
目的変数の分布

- 例えば、あるデータセットにはClass_A、Class_B、Class_Cという3クラスが存在するとします。
- クラスごとのレコード数が均等に分布していれば、分類モデルを作るの適したデータセットであるといえます。



目的変数の分布

- 機械学習は学習データ全体に対する誤差を最小化するアルゴリズムでモデルを作成するため、不均衡データ（クラス間に偏りがある）を使用すると、最も比率の多いクラス（下記であればClass_A）に最適化されたモデルが作成されてしまいます（レコード数が少ないクラスの判定精度が低下します）。



演習1：不均衡データの例

- 不均衡データの事例を挙げてください。

不均衡データの事例

クレジットカード不正利用検知
- 通常の使用回数に比べ不正利用される回数は極端に少ない。

機械の故障の早期検知
- 通常運転している時間の方が故障直前の時間より圧倒的に長い。

スパムメールの判定
- スパムメールのメール全体に占める割合は小さい。

演習2：不均衡データによるモデル作成

- 不均衡データで分類モデルを作成し、分類精度を確認してください。

補足：「不均衡＝質の悪いデータ」ではない

本講義の演習2を作成する際、当初はアヤメデータを題材に使用する予定でした。
アヤメデータにはsetosa/versicolor/virginicaの3つのクラスがあるため、人工的に不均衡データを作成してみました。
ですが、4つの説明変数sepal_length/sepal_width/petal_length/petal_widthのクラスに対する判別性能が高く、不均衡データを使用したモデルでも精度が非常に良いものとなりました。

不均衡データは一般的に質の悪いデータだと捉えられますが、よい説明変数を含んでいれば、一概に質が悪いとも言えないことがわかります。

興味のある方は「Chapter07-2_Imbalanced_data_iris.ipynb」を動かして上記の現象を確認してみてください。

第8回：不均衡データへの対応2

不均衡データへの対応

アジェンダ

- 第7回の振り返り
- 不均衡データへの対応

第7回の振り返り

データの加工：説明変数と目的変数の分離

- 通常、データセットは説明変数と目的変数が一体となっています。
- 機械学習モデルを作成する前に、どの列を目的変数として使用し、どの列を説明変数として使用するのかを決めます。

アヤメデータ

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。
あやめの種類を判定するための説明変数として使用する。

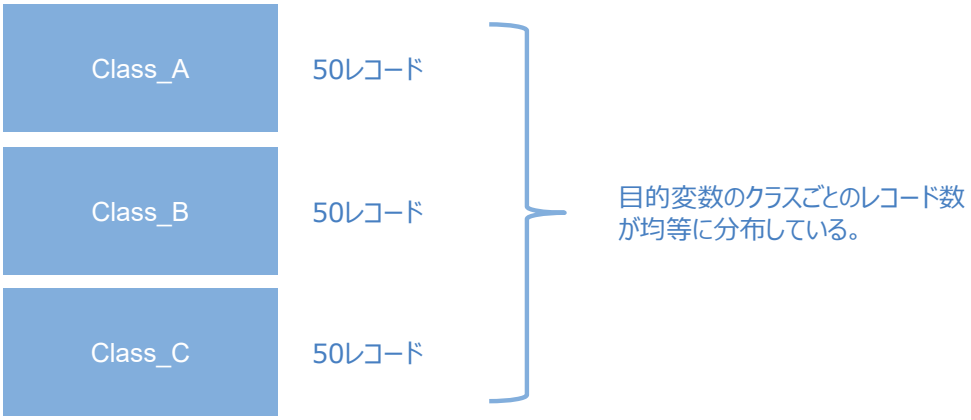
あやめの種類。
目的変数として使用する。

データの加工：学習データとテストデータの分離

- モデルの作成と評価を行うためには、データセットを4分割する必要があります。
- 学習データでモデルを作成し、テストデータでモデルの精度を評価します。

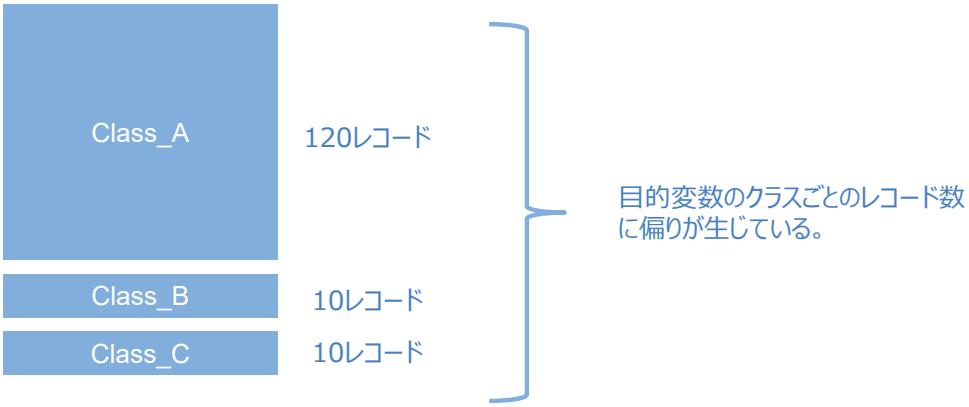
目的変数の分布

- 例えば、あるデータセットにはClass_A、Class_B、Class_Cという3クラスが存在するとします。
- クラスごとのレコード数が均等に分布していれば、分類モデルを作るの適したデータセットであるといえます。



目的変数の分布

- 機械学習は学習データ全体に対する誤差を最小化するアルゴリズムでモデルを作成するため、不均衡データ（クラス間に偏りがある）を使用すると、最も比率の多いクラス（下記であればClass_A）に最適化されたモデルが作成されてしまいます（レコード数が少ないクラスの判定精度が低下します）。



不均衡データの事例

クレジットカード不正利用検知
- 通常の使用回数に比べ不正利用される回数は極端に少ない。

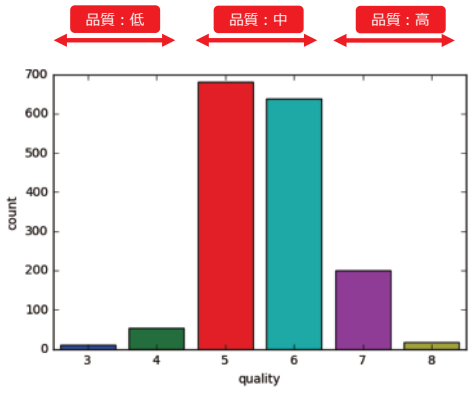
機械の故障の早期検知
- 通常運転している時間の方が故障直前の時間より圧倒的に長い。

スパムメールの判定
- スパムメールのメール全体に占める割合は小さい。

不均衡データへの対応

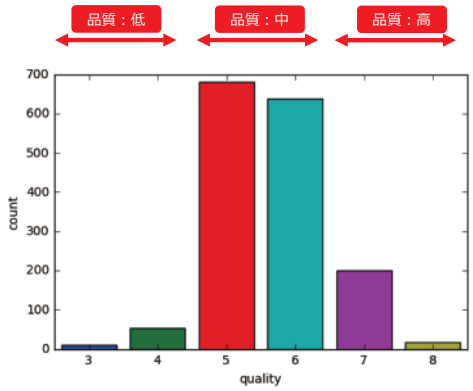
目的変数をどのように設定するのか

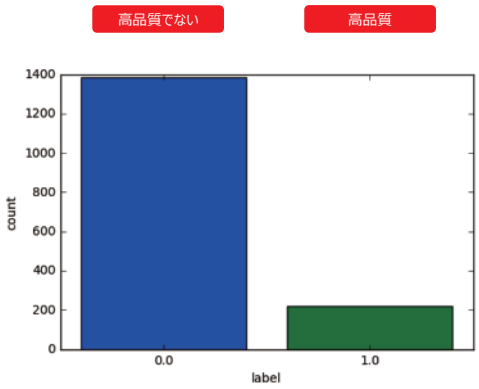
- 下記のグラフは、ワインに関するデータセット（https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv）のquality：品質という項目のヒストグラムです。
- 例えば高品質のワインを判定するモデルを作成したいのであれば、quality = 3~6と7~8を一括りにし、2クラスにまとめることが考えられます。



目的変数をどのように設定するのか

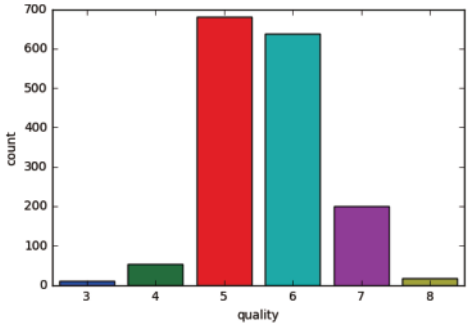
- 「高品質のワインを判定するモデルを作成したい」というように、モデルを作成する目的は何か？を考えれば、不均衡データとなってしまうのは仕方がありません。
- 「不均衡データにしたくないからquality:3~5と7~8で分けよう」としてしまうと、目的を達成できず本末転倒です。





演習1：生データの目的変数（ラベル）の確認

- 生データ（加工前のデータ）を観察し、どの項目を目的変数とするのか考えてください。
- 目的変数に含まれるクラスの数を確認してください。
- 目的変数のクラスを二値化してください。

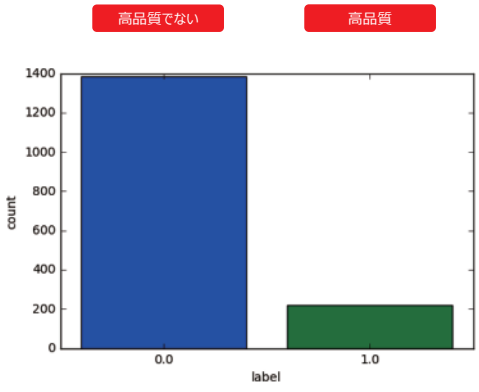


演習2：不均衡データによるモデル作成

- 不均衡データで分類モデルを作成し、分類精度を確認してください。

不均衡データへの対応：アンダーサンプリング

- 数が多い方のクラスのレコードをランダムにサンプリングして、数が少ないクラスのレコード数に合わせる手法です。

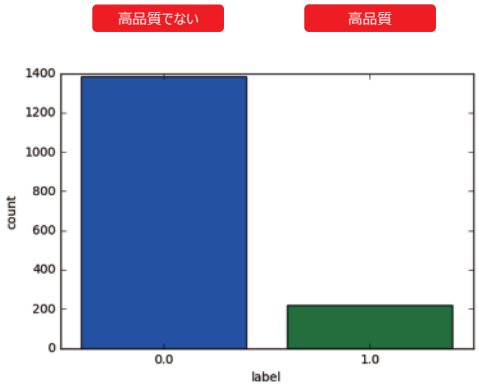


演習3：アンダーサンプリング

- アンダーサンプリングを実施し、モデルの精度を確認してください。

不均衡データへの対応：オーバーサンプリング

- 数が少ない方のクラスのレコードを、数が多いクラスのレコード数に合わせて増やす手法です。



演習4：オーバーサンプリング

- オーバーーサンプリングを実施し、モデルの精度を確認してください。

補足：アンダー/オーバーサンプリングの効果

ワインデータで実施した結果を右に示します。
モデルが高品質だと判定したワインのうち、本当に高品質であったワインの割合（Precision）が最も大きいのは不均衡データをそのまま使用したモデルでした。

また、高品質ワインのうち最も多くのワインを高品質だと判定できた割合（Recall）が最も大きいのは、アンダーサンプリングデータを使用したモデルでした。

オーバーサンプリングは、その中間の性能でした。

不均衡データそのまま

		実際		
		正例	負例	precision
予測	正例	44	29	60.3%
	負例	47	680	93.5%
	recall	48.4%	95.9%	90.5%

アンダーサンプリング

		実際		
		正例	負例	precision
予測	正例	70	153	31.4%
	負例	21	556	96.4%
	recall	76.9%	78.4%	78.3%

オーバーサンプリング

		実際		
		正例	負例	precision
予測	正例	53	46	53.5%
	負例	38	663	94.6%
	recall	58.2%	93.5%	89.5%

補足：アンダー/オーバーサンプリングの効果

アンダー/オーバーサンプリングを実施したからと言ってPrecision/Recallが両方とも格段に向上するわけではなく、ケース・バイ・ケースです（ときには全く効果がない場合もあります）。

データセットごとに様々な施策を実施し、モデルの精度向上を図ることが求められます。

不均衡データそのまま

		実際		
		正例	負例	precision
予測	正例	44	29	60.3%
	負例	47	680	93.5%
	recall	48.4%	95.9%	90.5%

アンダーサンプリング

		実際		
		正例	負例	precision
予測	正例	70	153	31.4%
	負例	21	556	96.4%
	recall	76.9%	78.4%	78.3%

オーバーサンプリング

		実際		
		正例	負例	precision
予測	正例	53	46	53.5%
	負例	38	663	94.6%
	recall	58.2%	93.5%	89.5%

第9回：自然言語処理

自然言語処理の概要

アジェンダ

- 自然言語処理とは
- 自然言語処理の技術
  - 単語分解
  - 構文解析
  - 意味解析
  - 文脈解析

自然言語処理とは

自然言語処理とは

「自然言語」とは何か（Wikipedia：https://ja.wikipedia.org/wiki/自然言語）

- 人間がお互いにコミュニケーションを行うための自然発生的な言語である。「自然言語」に対置される語に「形式言語」「人工言語」がある。形式言語との対比では、その構文や意味が明確に揺るぎなく定められ、利用者に厳格な規則の遵守を強いる（ことが多い）形式言語に対し、話者集団の社会的文脈に沿った曖昧な規則が存在していると考えられるものが自然言語である。自然言語には、規則が曖昧であるがゆえに、話者による規則の解釈の自由度が残されており、話者が直面した状況に応じて規則の解釈を変化させることで、状況を共有する他の話者とのコミュニケーションを継続する事が可能となっている。

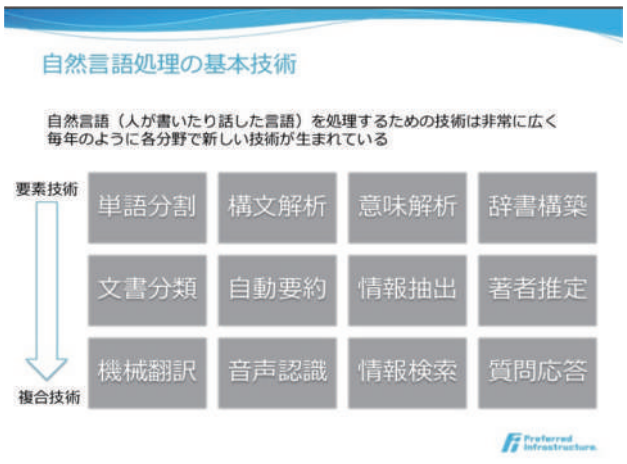
自然言語処理とは

- 「自然言語」は文化圏によって英語/日本語のように表記、文法、発音が異なります。
- 対峙している人同士、場面によって、共通認識を元にしたコミュニケーションが発生するため、全ての情報が自然言語に落ちているわけではなく、解釈が曖昧になることがあります。

- このように複雑な「自然言語」をコンピュータに処理させる技術群のことを自然言語処理といいます。

自然言語処理とは

- 自然言語処理は様々な技術で構成、整理されており、また応用分野も多岐に渡ります。



出展： https://www.slideshare.net/pfi/ss-11474303

質問

- 自然言語処理が応用されている分野について調べてください。
- 現時点で技術の詳細を理解する必要はありません。自然言語処理の技術でどのようなことができるのか、漠然としたイメージで捉える程度で結構です。

自然言語処理の技術

単語分割：形態素解析

形態素解析（Wikipedia：https://ja.wikipedia.org/wiki/形態素解析）

- 形態素解析（けいたいそかいせき、Morphological Analysis）とは、文法的な情報の注記の無い自然言語のテキストデータ（文）から、対象言語の文法や、辞書と呼ばれる単語の品詞等の情報にもとづき、形態素（Morpheme, おおまかにいえば、言語で意味を持つ最小単位）の列に分割し、それぞれの形態素の品詞等を判別する作業である。

「おまちしております。」を形態素解析した結果

文字列	読み	原形	品詞の種類	活用の種類	活用形
お待ち	オマチ	お待ち	名詞-サ変接続		
し	シ	する	動詞-自立	サ変・スル	連用形
て	テ	て	助詞-接続助詞		
おり	オリ	おる	動詞-非自立	五段・ラ行	連用形
ます	マス	ます	助動詞	特殊・マス	基本形
。	。	。	記号-句点		

単語分割：N-gram解析

N-gram解析（Wikipedia：https://ja.wikipedia.org/wiki/全文検索）

- 検索対象を単語単位ではなく文字単位で分解し、後続の N-1 文字を含めた状態で出現頻度を求める方法。Nの値が1なら「ユニグラム（英: uni-gram）」、2なら「バイグラム（英: bi-gram）」、3なら「トライグラム（英: tri-gram）」と呼ばれる。たとえば「全文検索技術」という文字列の場合、「全文」「文検」「検索」「索技」「技術」「術（終端）」と2文字ずつ分割して索引化を行ってやれば、検索漏れが生じず、辞書の必要も無い。

- 「全文検索技術」という文字列をバイグラムすると「全文」「文検」「検索」「索技」「技術」「術（終端）」と2文字ずつ分割されます。

単語分割：形態素解析とN-gram解析の比較

- N-gramには辞書が不要、という利点はありますが、ノイズが大きいという短所があります（例えば、「東京都」をバイグラムで分割した場合、「東京」と「京都」という結果が含まれてしまいます。）。
- もし検索エンジンにN-gramを採用すると、「東京都の天気」が知りたいのに「東京の天気」と「京都の天気」を検索してしまうかもしれません。
- 機械学習の前処理として形態素解析、N-gram解析を使用する際は、両者の特性を考慮した選択が必要です。

形態素解析とN-gramの比較

	形態素解析	N-gram
インデクシング速度	遅い	速い
インデックスサイズ	小さい	大きい
検索ノイズ	少ない	多い
検索漏れ	多い	少ない
検索速度	速い	遅い
言語依存	辞書が必要	辞書が不要

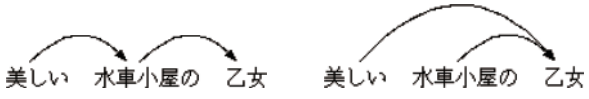
出展： https://ja.wikipedia.org/wiki/全文検索

構文解析

構文解析（Wikipedia：https://ja.wikipedia.org/wiki/構文解析）

- 構文解析（こうぶんかいせき、syntactic analysis あるいは parse）とは、文章、具体的にはマークアップなどの注記の入っていないベタの文字列を、自然言語であれば形態素に切分け、さらにその間の関連（修飾-被修飾など）といったような、統語論的（構文論的）な関係を図式化するなどして明確にする（解析する）手続きである。

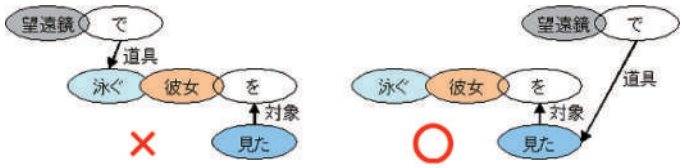
- 「美しい 水車小屋の 乙女」という文章には少なくとも2つの解釈が存在する。「水車小屋が美しい」場合と、「乙女が美しい」場合である。この場合には、意味を含めても正しい解釈がどちらであるか不明であり、その文が置かれた前後の状況、言い換えるとコンテキスト、フレーム情報などを考慮しなければ同定できない。



意味解析

- 下記の例は「望遠鏡で泳ぐ彼女を見た」という文章の解析例です。
- 人間であれば、「望遠鏡を使って泳ぐ人はいない」と判断できるため、すぐに右が正しいと分かります。
- このように、構文解析の結果が「意味」として適当かを判断することを意味解析といいます。

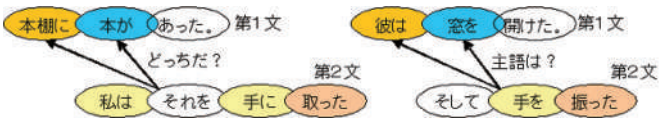
- その他の問題として、「多義性解消」というものがあります。
- 例えば「やった」という言葉には、「宿題をやった（実施した）」のような場合と、「本をやった（与えた）」という解釈の仕方があります。これをどちらか判断することも意味解析といいます。



出展： http://www.sist.ac.jp/~kanakubo/research/natural_language_processing.html

文脈解析

- 構文解析が文単位で行なわれるのに対し、複数の文にまたがる構文木作成＋意味解析を行なうのが文脈解析です。文脈解析は長い文脈に即して行なう必要があるため、単独文の意味解析よりはさらに複雑となります。
- 例えば、「それ」という代名詞が指すのは何か、という問題は文脈解析で解決します。



出展： http://www.sist.ac.jp/~kanakubo/research/natural_language_processing.html

質問

- これまで単語分割、構文解析、意味解析、文脈解析の概要を学んできました。
- 皆さんが普段話す会話を分析する際、どの技術を使えば何ができるのか議論してください。

第10回：言語資源

言語資源の種類と役割

アジェンダ

- 言語資源とは
- 言語資源の種類と役割
  - 辞書
  - コーパス

言語資源とは

言語資源とは

言語資源（Wikipedia：https://ja.wikipedia.org/wiki/言語資源）

- 言語資源（げんごしげん）とは、自然言語を研究するさいに用いられる資源のこと。辞書やコーパス、シソーラス、インフォーマントなどがこれにあたる。電子化された言語資源は自然言語処理技術の研究に不可欠であるが、作成に非常に手間がかかるため、いまだにその数は少なく、一般にとても高価である。

言語資源の種類

言語資源は辞書とコーパスに大別されます。両者の違いは、取り扱う粒度の違いです。

辞書
- 特定の言語単位（音素、形態素、単語など）に対する言語情報資源のことです。
- ただの単語集合のみであったり、属性情報（品詞等）が付与されていたりします。

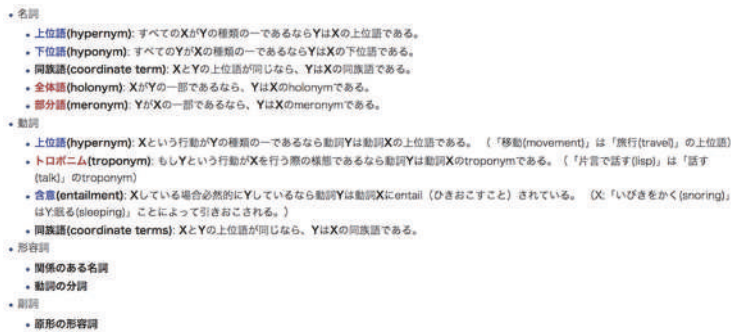
コーパス
- 自然言語の文章を構造化し大規模に集積したものです。
- 構造化し、言語的な情報（品詞、統語構造など）を付与しています。

辞書の例：IPA辞書

- 有名な日本語用辞書としてIPA辞書があります。
- IPA品詞体系（http://chasen.naist.jp/snapshot/ipadic/ipadic/doc/ipadic-ja.pdf）を元に作成された辞書です。

辞書（シソーラス）の例：WordNet

- 概念の辞書です。
- 対象単語の上位、下位、同義語など、概念の対象関係を辞書にしたものです。



- 名詞
  - 上位語(hypernym): すべてのXがYの種類の一つであるならYはXの上位語である。
  - 下位語(hyponym): すべてのYがXの種類の一つであるならYはXの下位語である。
  - 同族語(coordinate term): XとYの上位語が同じなら、YはXの同族語である。
  - 全体語(holonym): XがYの一部であるなら、YはXのholonymである。
  - 部分語(meronym): YがXの一部であるなら、YはXのmeronymである。
- 動詞
  - 上位語(hypernym): Xという行動がYの種類の一つであるなら動詞Yは動詞Xの上位語である。（「移動(movement)」は「旅行(travel)」の上位語）
  - トロポニム(troponym): もしYという行動がXを行う際の様態であるなら動詞Yは動詞Xのtroponymである。（「片言で話す(lisp)」は「話す(talk)」のtroponym）
  - 含意(entailment): Xしている場合必然的にYしているなら動詞Yは動詞Xにentail（ひきおこすこと）されている。（X:「いびきをかく(snoring)」はY:眠る(sleeping)」ことによって引きおこされる。）
  - 同族語(coordinate terms): XとYの上位語が同じなら、YはXの同族語である。
- 形容詞
  - 関係のある名詞
  - 動詞の分詞
- 副詞
  - 原形の形容詞

出展： https://ja.wikipedia.org/wiki/WordNet

コーパスの例：京都大学テキストコーパス

京都大学テキストコーパス Version 4.0（http://shachi.org/resources/4227）

- 毎日新聞95年1月1日から17日までの全記事（約2万文）、1月から12月までの社説記事（約2万文）、計約4万文に対して京都大学の形態素解析システム(JUMAN)、構文解析システム(KNP)で自動解析を行い、その結果を人手修正したテキストコーパス。

```
README "京都大学テキストコーパス Version 4.0 2005/04/22
格関係、照応・省略関係、共参照情報付きの場合：

# S-ID:950101001-001
* 0 2D
+ 0 3D
太郎 たろう * 名詞 人名 * *
は は * 助詞 副助詞 * *
* 1 2D
+ 1 2D
東京 とうきょう * 名詞 固有名詞 * *
+ 2 3D
大学 だいがく * 名詞 普通名詞 * *
に に * 助詞 格助詞 * *
* 2 -1D
+ 3 -1D
行った いった 行く 動詞 * 子音動詞カ行促音便形 基本形
EOS
```

言語資源へのアクセス

形態素解析

- 形態素解析器はいくつかありMeCabが有名ですが、本講義ではpythonから簡単に使用できるJanome（http://mocobeta.github.io/janome/#id13）を紹介します。

- Janome (蛇の目) は，Pure Python で書かれた，辞書内包の形態素解析器です。
- 依存ライブラリなしで簡単にインストールでき，アプリケーションに組み込みやすいシンプルな API を備える形態素解析ライブラリを目指しています。
- 内包辞書として mecab-ipadic-2.7.0-20070801 を使っています。

演習1：形態素解析実行環境の構築

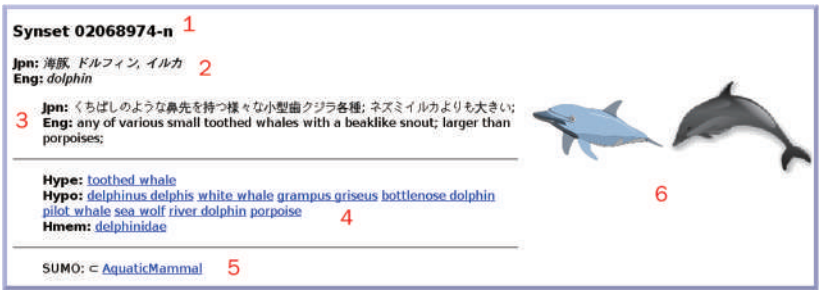
- 次のコマンドを実施し、python環境にJanomeをインストールしてください。

演習2：形態素解析の実行

- 実際に形態素解析を実行し、結果を確認してください。

日本語WordNet

- 国立研究開発法人情報通信研究機構（NICT）が構築した日本語のWordNet（ http://compling.hss.ntu.edu.sg/wnja/ ）です。



演習3：WordNet参照環境の構築

- 日本語WordNetの参照環境を構築してください。
- データは[resources/wnjpn.db.gz]に格納しています。解凍して使用してください。

演習4：類義語の検索

- WordNetで類義語を検索してください。

質問

- 本講義で形態素解析と概念辞書の使い方を学びました。これらの技術を組み合わせるとどのようなサービスが開発可能か、アイディアを出し合ってみてください。

第11回：自然言語処理の前処理

自然言語のベクトル化

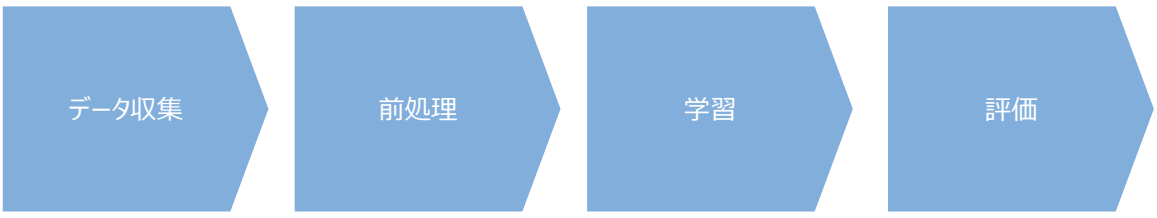
アジェンダ

- 第1回講義（データ前処理）の振り返り
- 自然言語処理における前処理

第1回講義（データの前処理）の振り返り

データの前処理とは

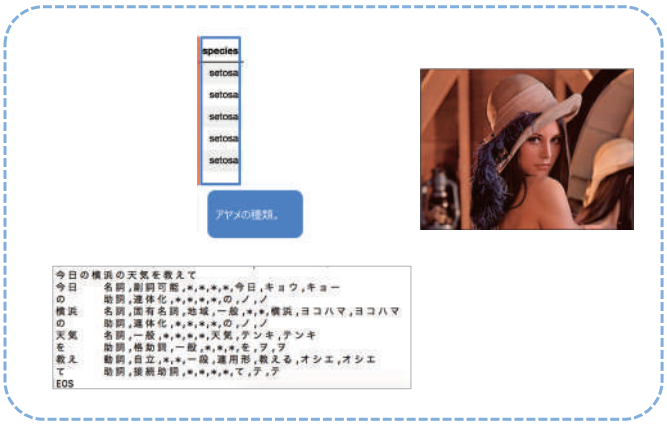
- 「学習」ステップにおいて機械学習などを活用してモデルを作成しますが、その前ステップとして「前処理」が必要となります。
- 前処理は、機械学習器を使用するためには必須である前処理と、モデル精度を向上させるために実施することが望ましい前処理とに大別できます。



モデル作成のステップ

必須の前処理とは

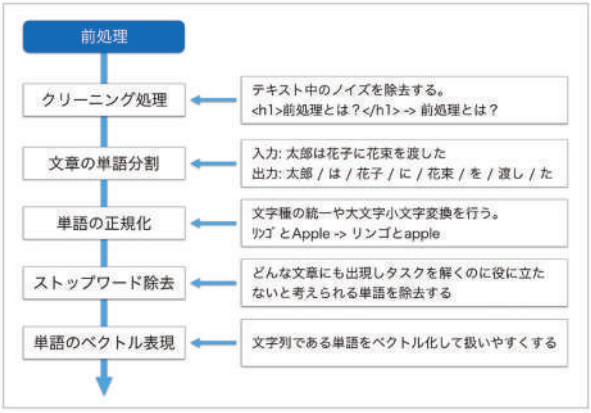
- 機械学習器は数値を扱うことができます。区分値や自然言語、画像などは機械学習器で扱える数値データに変換する必要があります。
- 例え数値データを扱うとしても、欠損値が存在する場合はそのままでは機械学習器に掛ける事ができない場合があります。



自然言語処理における前処理

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。

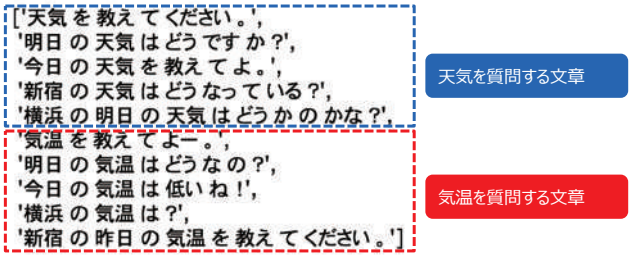
```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 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)
```

演習1：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2：クラスタリングの実施

- ベクトル化した文章をクラスタリングしてください。
- クラスタリング結果は、「天気を質問する文章」と「気温を質問する文章」をどの程度分類できているのか観察してください。



文章のベクトル化：TF-IDF

- 文章をベクトル化する際に、単語に重みを付けて評価する手法です。単語の出現頻度であるTF（Term Frequency）と、IDF（Inverse Document Frequency）という2つの指標を使用します。
- TF（Term Frequency）は、「各文書においてその単語がどのくらい出現したのか」を意味します。よく出現する単語は、その文章の特徴を捉えるのに有用だろうという考え方です。
- IDF（Inverse Document Frequency）は、単語が稀にしか出現しないなら高い値を、「色々な文書によく出現する単語」なら低い値を示すものです。稀少な単語は、その文書の特徴を捉えるのに有用だろうという考え方です。

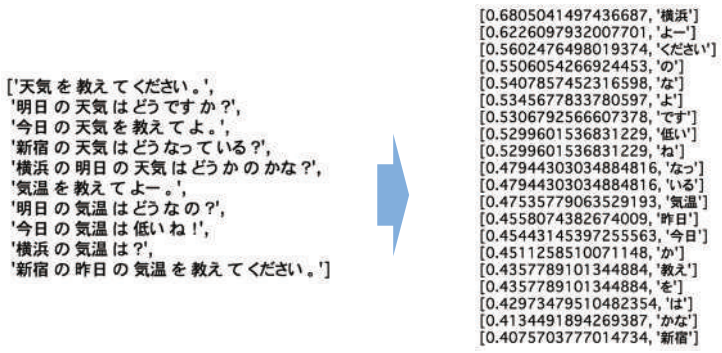
$$tf = \frac{\text{文書}A\text{における単語}X\text{の出現頻度}}{\text{文書}A\text{における全単語の出現頻度の和}} \qquad idf = log(\frac{\text{全文書数}}{\text{単語}X\text{を含む文書数}}) \qquad tfidf = tf * idf$$

演習3：TF-IDFの実装

- TF-IDFを実装し、作成したベクトルを確認してください。

演習4：TF-IDFによる重要単語の判定

- TF-IDFで作成したベクトルの、スコアの高い形態素の上位N件を表示するプログラムを作成してください。

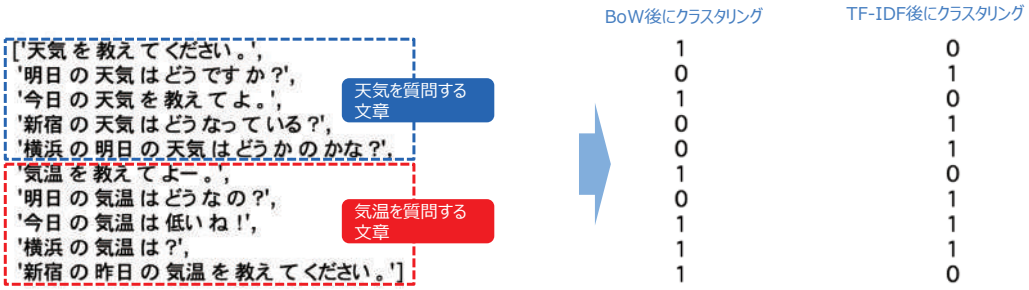


演習5：クラスタリングの実施

- TF-IDFで作成した新しいベクトルで再度クラスタリングしてください。

ストップワードを設定しなかった例

- ストップワード（自然言語処理する際に役に立たないため処理対象外とする単語のことです。例えば助詞や助動詞などの機能語（「は」「の」「です」「ます」など）が挙げられます。）を除去せずにベクトル化した文章でクラスタリング（K-Means、K=2の結果）した例です。
- 上の5文と下の5文でグルーピングされてほしかったのですが、そうなっていません。特に「天気を質問する文章」が「気温を質問する文章」と判断されていることがわかります。
- 数値データを扱う機械学習と同じで、特徴をよく捉えたベクトルを作成することができなければ、良い結果が得られないという事例です。



演習6： BoWの実装（ストップワードの除去）

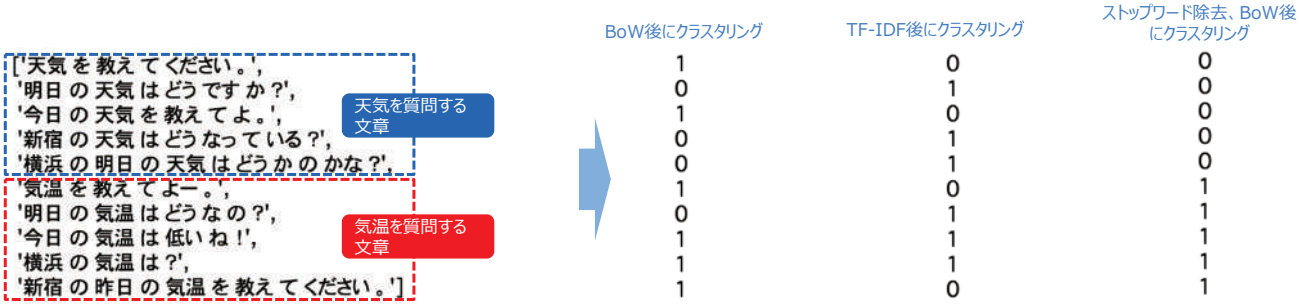
- 名詞と動詞のみを使用して、再度ベクトルを作成してください。

演習7：クラスタリングの実施

- 名詞と動詞のみで作成したベクトルでクラスタリングを実施してください。

ストップワードを設定した例

- ストップワード除去後にクラスタリングした例（一番右）です。文章が理想的にクラスタリングされていることが確認できます。



第12回：自然言語データ収集・抽出

スクレイピングの手法

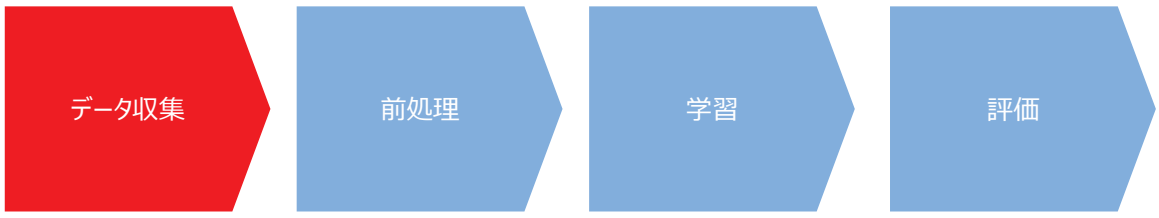
アジェンダ

- データ収集・抽出が必要となる場面
- データ収集・抽出の実践
- データ収集の戦略

なぜデータ収集が必要となるのか

データ収集・抽出が必要となる場面

- 何かしらのモデルを作成するためにはデータが必要です。公開されているコーパスを使用してモデルを作成することは可能です。
- 公開されているコーパスだけでは不十分な場合があります。例えば、コーパスがカバーしている単語、言い回しでは目的のモデルを作成するには量・質が足りない、という場合です。

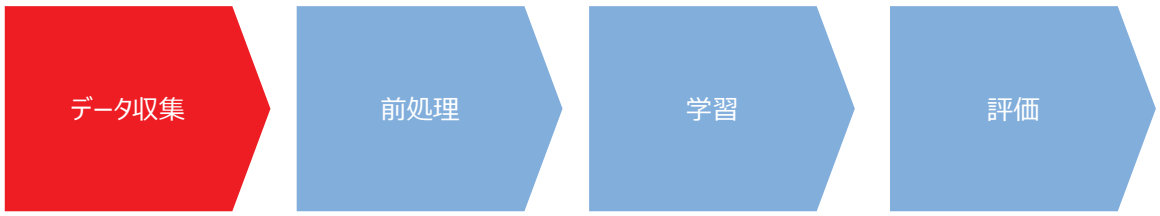


モデル作成のステップ

データ収集・抽出が必要となる場面

独自にデータを収集しなければならない事例には、下記の様なものがあります。

- 個人情報を含む文章でモデルを作成する必要があるが、関係者以外に個人情報を表示したくないため、対象外リストを独自に構築しなければならない。
- 金融・法律分野で使用される文章を解析したいので、専門用語を認識できるモデルを作成しなければならない。
- 若者言葉、微妙なニュアンスの文章の解析をしたいので、公開されているコーパスのみでは単語・表現ともにバリエーションが足りない。



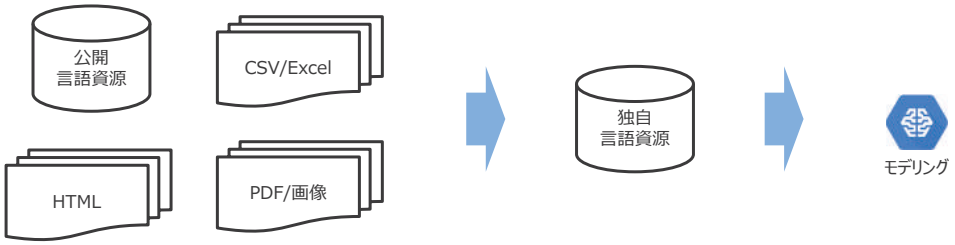
モデル作成のステップ

質問：独自データ収集

- 前ページで挙げた事例以外に、独自データを収集する必要があるのはどのような場面か、議論してください。

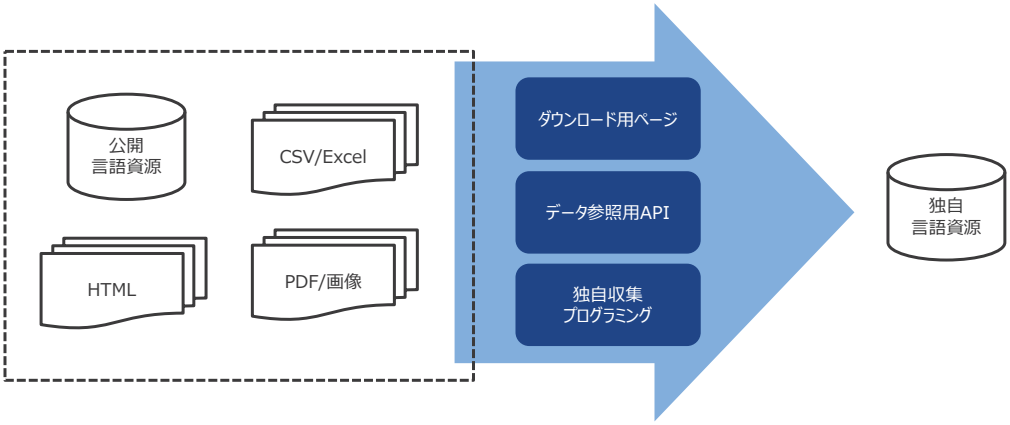
収集するデータの形式

- 収集対象のデータ形式は多岐に渡ります。
- 既に使いやすい形式（データベース、CSVなど）になっていることは稀で、モデリングに使いたい文字列のみをうまく抽出する必要があります。



収集するデータの形式

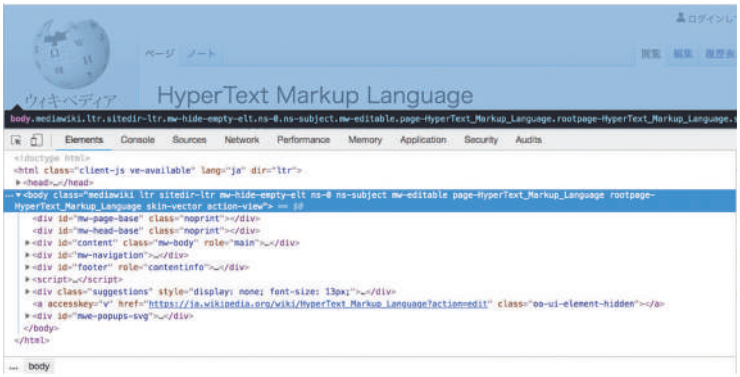
- インターネットを経由してデータを収集する場合、ダウンロード専用のページやAPIが公開されていれば、あまりコストを掛けずにデータを収集することができます。
- 収集の手段が用意されていなければ、独自に収集の仕組みを構築する必要があります。



データ収集・抽出の実践

HTMLのスクレイピング

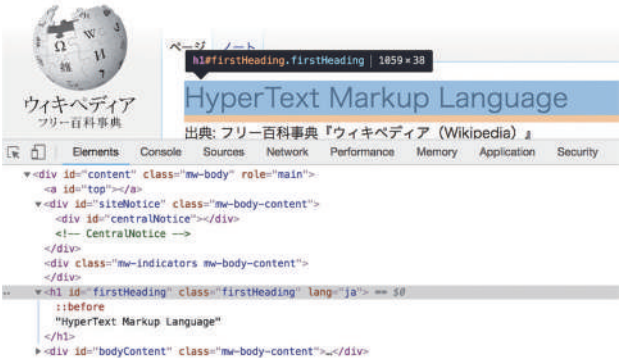
- HTMLとは、Hyper Text Markup Language（ハイパーテキスト・マークアップ・ランゲージ）の略で、Webページを作るための最も基本的なマークアップ言語のひとつです。
- 普段、私たちがブラウザで観ているWebページのほとんどが、HTMLで作られています。



Google ChromeからWikipediaのページをHTMLで表示した例

HTMLのスクレイピング

- 例えばWikipediaのページ（ https://ja.wikipedia.org/wiki/HyperText_Markup_Language ）の見出しから「HyperText Markup Language」という文字列を抽出するためには、HTMLのヘッダ部分にある<h1>タグで挟まれている部分を見つけ出して、切り出す必要があります。このような操作をスクレイピングといいます。



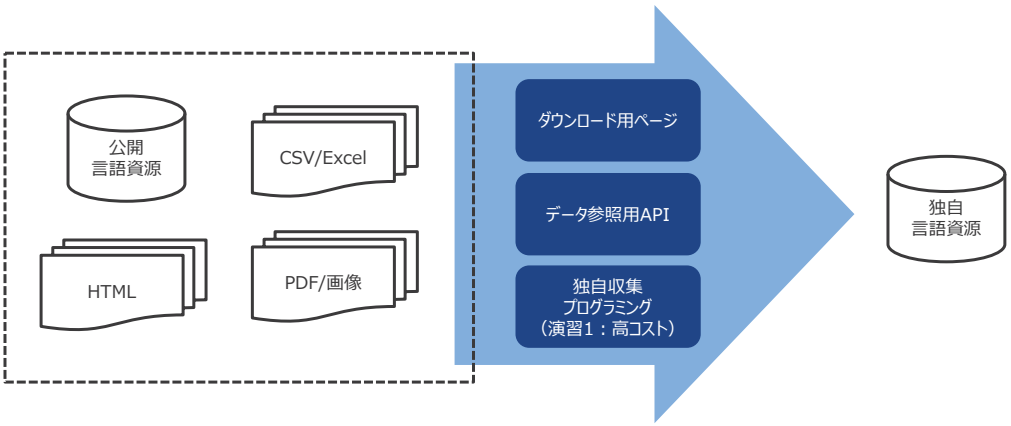
Google ChromeからWikipediaのページをHTMLで表示した例

演習1：Wikipediaのスクレイピング

- Wikipediaの任意のページから文章を抽出するプログラムを実装してください。

収集するデータの形式

- 演習1においてWikipediaのページをスクレイピングしました。HTMLを手元に落とし、中身を確認してタグを除去してと、非常に手間がかかることがわかります。
- 実はWikipediにはAPIが存在し、はるかに低コストで文字列を取得することができます。

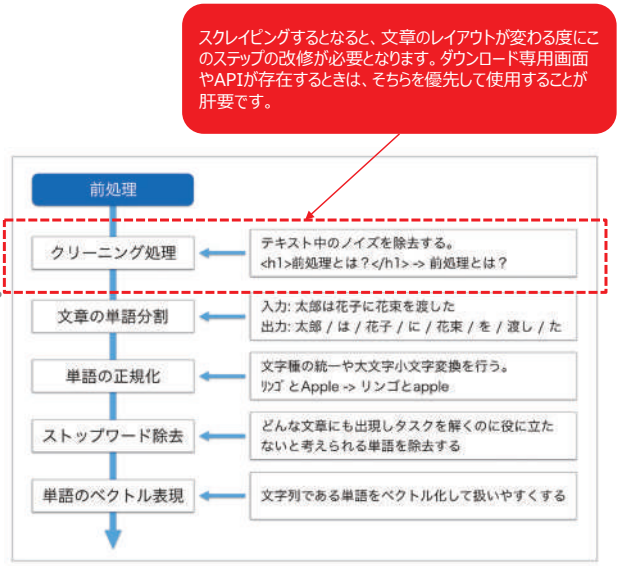


演習2：WikipediaへのAPIアクセス

- APIを使用してWikipediaの任意のページから文章を抽出するプログラムを実装してください。

データ収集の戦略

- 演習1と演習2でデータの取得コストが大きく違うことを学習しました。
- 機械学習モデルを作成するためには大量のデータを必要とするため、データ収集の仕組みはそれ自体で1つのシステムとなるほどの規模です。
- 取得対象とするデータは何なのか、どのような形式で取得できるのか、予め計画を立ててプロジェクトを進めなければ、データ収集コストが肥大するのを避けられなくなってしまいます。



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

データ収集の戦略

- データの取得にコストが掛かるということを学習しました。また、機械学習モデルを構築するためには大量のデータが必要だということも学習しました。
- データを使う側の立場としては、APIを駆使して低コストでデータを収集することが肝要ですが、APIが存在しないデータをあえてスクレイピングで取得することがあります。

- 繰り返しになりますが、機械学習モデルの構築にはデータが必要です。そのデータは、簡単に取得できないものでかつ需要があるほど高価になります。モデリングする企業に対してデータを売却することをビジネスとする企業が存在するほど、ビジネスとしてはホットな分野なのです。

第13回：画像処理の前処理

画像のベクトル化

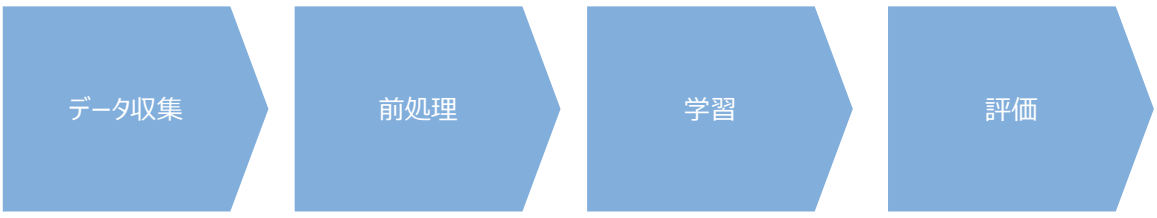
アジェンダ

- 第1回講義（データ前処理）の振り返り
- 画像処理における前処理

第1回講義（データの前処理）の振り返り

データの前処理とは

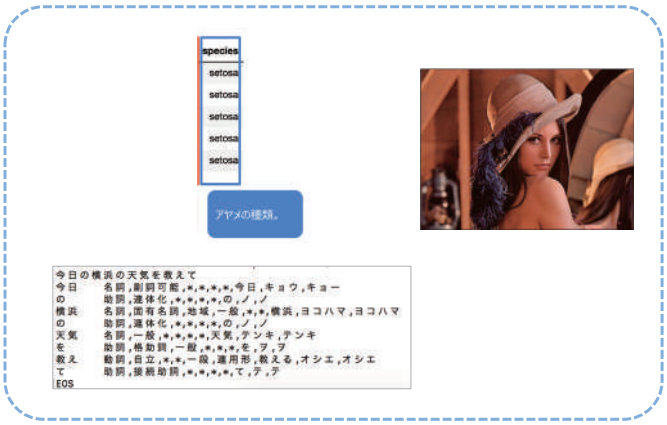
- 「学習」ステップにおいて機械学習などを活用してモデルを作成しますが、その前ステップとして「前処理」が必要となります。
- 前処理は、機械学習器を使用するためには必須である前処理と、モデル精度を向上させるために実施することが望ましい前処理とに大別できます。



モデル作成のステップ

必須の前処理とは

- 機械学習器は数値を扱うことができます。区分値や自然言語、画像などは機械学習器で扱える数値データに変換する必要があります。
- 例え数値データを扱うとしても、欠損値が存在する場合はそのままでは機械学習器に掛ける事ができない場合があります。

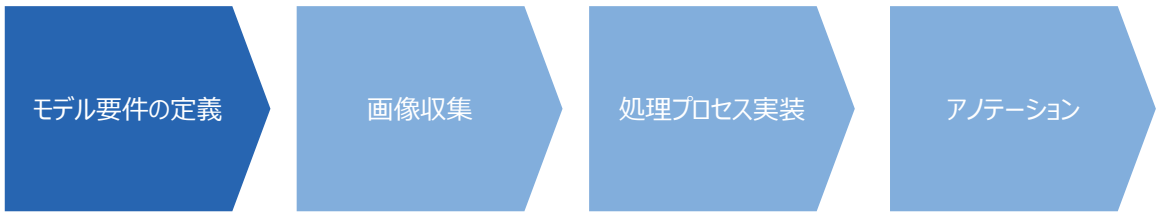


数値データ

画像処理における前処理

モデル要件の定義

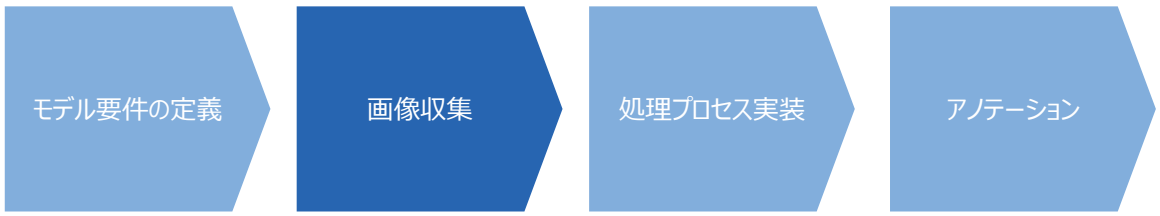
- これから作成するモデルは何の画像を判定できるようにしたいのか、をまずは決定します。
- 例えばコンビニ各社の看板を判定できるモデルを作りたい場合、「セブンイレブン、ローソン、ファミリーマートの3社の看板を判定したい」、と定義します。



画像処理におけるデータ作成のプロセス

画像収集

- モデルで判定したい対象の画像を収集します。



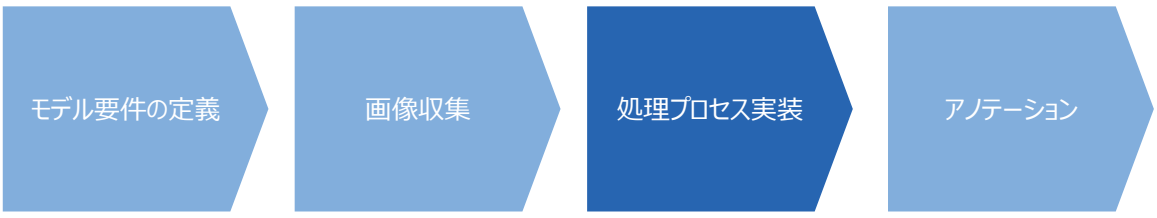
画像処理におけるデータ作成のプロセス

演習1：画像の収集

- セブンイレブン、ローソン、ファミリーマートの看板の写真を収集してください。

処理プロセス実装

- 画像のサイズを統一します。
- 画像処理ライブラリPillowなどでサイズ変換処理が実装できます。



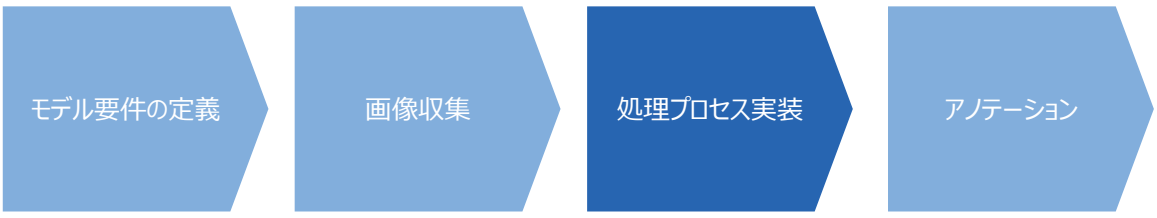
画像処理におけるデータ作成のプロセス

演習2：画像のサイズ変更

- 収集したセブンイレブン、ローソン、ファミリーマートの看板の写真サイズを、64ピクセル×64ピクセルに変換してください。

処理プロセス実装

- 画像の左右反転、回転処理を行い、画像データの水増しを行います。



画像処理におけるデータ作成のプロセス

演習3：画像の左右反転

- 収集したセブンイレブン、ローソン、ファミリーマートの看板の写真に左右反転を施してください。

演習4：画像の回転

- 収集したセブンイレブン、ローソン、ファミリーマートの看板の写真に90度、180度、270度回転を施してください。

なぜデータの水増しを行うか

- 画像には判定に寄与する領域と、ノイズになる領域があります。
- 判定に寄与する領域が、常に画像の真ん中に位置するわけではなく、また常に方向が一定というわけでもありません。

- なるべく学習データのパターンを増やしロバストなモデルとするために、画像データの水増しは有効な手段です。

演習5：CNNによるモデル作成

- CNNでコンビニの看板を判定するモデルを作成してください。

Kerasの便利な関数について

- 今回の講義、演習では、画像の処理にどのようなものがあるのかを学習するために画像リサイズ、左右反転、回転処理を実際にプログラムを実装して実施してきました。
- これらの処理は画像認識モデルを作成する際にはほぼ必須の処理であるため、予めKerasで便利な関数（https://keras.io/ja/preprocessing/image/）が準備されています。

参考サイト
http://pynote.hatenablog.com/entry/keras-image-data-generator
https://qiita.com/yoyoyo_/items/0034e5e82813b05e41df

第14回：openCVによる画像処理1

アジェンダ

- openCVとは
- openCVで実施できる処理

openCVとは

openCVとは

openCV（出展：https://ja.wikipedia.org/wiki/OpenCV）

- OpenCV（オープンシーヴィ、英語: Open Source Computer Vision Library）とはインテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリ。2009年にWillow Garage（ウィロー・ガレージ）に開発が移管された後、2015年現在はItseezがメンテナンスを行なっている。なお、2016年5月26日にインテルがItseezを買収することが発表された。

コンピュータビジョン（出展：https://ja.wikipedia.org/wiki/コンピュータビジョン）

- コンピュータビジョン（computer vision）はコンピュータがデジタルな画像、または動画をいかによく理解できるか、ということを扱う研究分野である。工学的には、人間の視覚システムが行うことができるタスクを自動化することを追求する分野である。

openCVとは

OpenCVは下記のような関数を搭載しており、ビジョンの研究者や開発者の仕事を楽にするためのツールです。

- 線形代数や統計処理など，コンピュータビジョンに必要な各種数学関数
- 直線や曲線，テキストなど画像への描画関数
- OpenCVで使用したデータを読み込み/保存するための関数
- エッジ等の特徴抽出や画像の幾何変換，カラー処理等々の画像処理関数
- 物体追跡や動き推定などの動画像処理用関数
- 物体検出などのパターン認識関数
- 三次元復元のためのカメラ位置や姿勢の検出などのカメラキャリブレーション関数
- コンピュータにパターンを学習させるための機械学習関数
- 画像の読み込みや保存，表示，ビデオ入出力などインターフェース用関数

openCVで実施できる処理

画像データの読み込み

- 画像を読み込むと3次元配列がピクセル分連なった行列に変換されます。
- 3次元配列の中身はBGRの順番となっています。

```
# 読み込みたい画像のパスを指定します。
path_img = 'resources/images/books.jpg'

"""
画像を読み込みます。読み込むと数値に変換されます。
3次元配列(BGR)がピクセル数連なった行列に変換されます。
"""
img = cv2.imread(path_img)

# 行列の中身を確認します。
print(img)
```

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

```
[[[160 157 153]
  [158 155 150]
  [153 151 143]
  ...
  [133 161 162]
  [129 160 163]
  [129 163 163]]

 [[158 157 153]
  [155 155 149]
  [150 151 142]
  ...
```

演習1：画像データの読み込み

- 画像を読み込んで数値に変換し、数値データを確認してください。

```
# 読み込みたい画像のパスを指定します。
path_img = 'resources/images/books.jpg'

"""
画像を読み込みます。読み込むと数値に変換されます。
3次元配列(BGR)がピクセル数連なった行列に変換されます。
"""
img = cv2.imread(path_img)

# 行列の中身を確認します。
print(img)
```

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

```
[[[160 157 153]
  [158 155 150]
  [153 151 143]
  ...
  [133 161 162]
  [129 160 163]
  [129 163 163]]

 [[158 157 153]
  [155 155 149]
  [150 151 142]
  ...
```

色の要素の順番変更

- OpenCVでは画像はBGRフォーマットで読み込まれる初期設定となっていますが、Matplotlibの場合はRGBで読み込まれます。R：赤とB：青の数値が入れ替わっているまま表示させると、下図のように色調が異なっていることが確認できます。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

matplotlibによって「BGR」で表示

matplotlibによって「RGB」で表示

演習2：色の要素の順番変更

- 画像フォーマットをBGRからRGBに変換し、変換前後で写真の見え方がどのように異なるか確認してください。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

matplotlibによって「BGR」で表示

matplotlibによって「RGB」で表示

グレースケール化

- グレースケール画像とは、最も暗い色（黒）を表す数値を0、また最も明るい色（白）を表す数値を255とし、ピクセルの明るさを表現する256階調のスケールで、1つの色チャンネルしか持っていない画像のことです。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

グレースケール化した画像

演習3：画像のグレースケール化

- 画像をグレースケールに変換し、数値データを確認してください。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

グレースケール化した画像

閾値処理

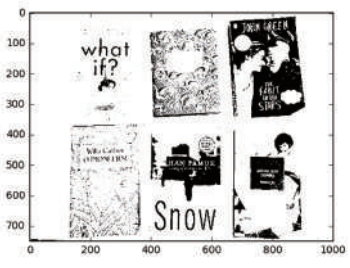
- ピクセルのチャンネル値がある閾値を超えた場合は、画像の各ピクセルを白いピクセルに、超えなかった場合は、黒いピクセルに置き換えます。
- 画像を2値画像、つまりシングルチャンネル画像に変換する処理で、画像セグメンテーションの一番簡単な方法です。
- グレースケール化はシングルチャンネル画像の一種です。

演習4：閾値処理

- グレースケール化した画像に閾値処理を実施してください。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

グレースケール化した画像



閾値処理を実施した画像

画像のフィルタリング処理：畳み込み

- 学習データやテストデータにノイズ除去や画像のぼかしを入れると、汎化性能の高いロバスト（頑健な）なモデル作成や判定を行うことができます。
- openCVでは入力画像とカーネル(フィルタ)のconvolution（畳み込み）を計算できます。以下に5x5サイズの平均値フィルタに使うカーネルを示します。

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

出展：http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering

演習5：画像の畳み込み

- 画像の畳込み処理を実行してください。
- フィルタは下図のように5×5のサイズとします。

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

出展：http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering

画像のフィルタリング処理：ガウシアンフィルタ

ガウシアンフィルタ（出展：http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering）

- 注目画素との距離に応じて重みを変えるガウシアンカーネルをフィルタに使用する事もできます。
- カーネルの縦幅と横幅(どちらも奇数)に加え、ガウシアンの標準偏差値sigmaX(横方向)とsigmaY(縦方向)を指定する必要があります。sigmaXしか指定されなければ、sigmaYはsigmaXと同じだとみなされます。どちらの値も0にした場合、カーネルのサイズから自動的に計算されます。
- ガウシアンフィルタは白色雑音の除去に適しています。

演習6：ガウシアンフィルタ

- ガウシアンフィルタによる画像の平滑化を実施してください。

第15回：openCVによる画像処理2

アジェンダ

- 第14回の振り返り
- openCVで実施できる物体抽出、輪郭抽出、形状補正処理

第14回の振り返り

openCVとは

openCV（出展：https://ja.wikipedia.org/wiki/OpenCV）

- OpenCV（オープンシーヴィ、英語: Open Source Computer Vision Library）とはインテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリ。2009年にWillow Garage（ウィロー・ガレージ）に開発が移管された後、2015年現在はItseezがメンテナンスを行なっている。なお、2016年5月26日にインテルがItseezを買収することが発表された。

コンピュータビジョン（出展：https://ja.wikipedia.org/wiki/コンピュータビジョン）

- コンピュータビジョン（computer vision）はコンピュータがデジタルな画像、または動画をいかによく理解できるか、ということを扱う研究分野である。工学的には、人間の視覚システムが行うことができるタスクを自動化することを追求する分野である。

openCVとは

OpenCVは下記のような関数を搭載しており、ビジョンの研究者や開発者の仕事を楽にするためのツールです。

- 線形代数や統計処理など，コンピュータビジョンに必要な各種数学関数
- 直線や曲線，テキストなど画像への描画関数
- OpenCVで使用したデータを読み込み/保存するための関数
- エッジ等の特徴抽出や画像の幾何変換，カラー処理等々の画像処理関数
- 物体追跡や動き推定などの動画像処理用関数
- 物体検出などのパターン認識関数
- 三次元復元のためのカメラ位置や姿勢の検出などのカメラキャリブレーション関数
- コンピュータにパターンを学習させるための機械学習関数
- 画像の読み込みや保存，表示，ビデオ入出力などインターフェース用関数

画像データの読み込み

- 画像を読み込むと3次元配列がピクセル分連なった行列に変換されます。
- 3次元配列の中身はBGRの順番となっています。

```
# 読み込みたい画像のパスを指定します。
path_img = 'resources/images/books.jpg'

"""
画像を読み込みます。読み込むと数値に変換されます。
3次元配列(BGR)がピクセル数連なった行列に変換されます。
"""
img = cv2.imread(path_img)

# 行列の中身を確認します。
print(img)
```

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

```
[[[160 157 153]
  [158 155 150]
  [153 151 143]
  ...
  [133 161 162]
  [129 160 163]
  [129 163 163]]

 [[158 157 153]
  [155 155 149]
  [150 151 142]
  ...
```

色の要素の順番変更

- OpenCVでは画像はBGRフォーマットで読み込まれる初期設定となっていますが、Matplotlibの場合はRGBで読み込まれます。R：赤とB：青の数値が入れ替わっているまま表示させると、下図のように色調が異なっていることが確認できます。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

matplotlibによって「BGR」で表示

matplotlibによって「RGB」で表示

グレースケール化

- グレースケール画像とは、最も暗い色（黒）を表す数値を0、また最も明るい色（白）を表す数値を255とし、ピクセルの明るさを表現する256階調のスケールで、1つの色チャンネルしか持っていない画像のことです。

出展：https://github.com/piratefsh/image-processing-101/tree/master/images

グレースケール化した画像

閾値処理

- ピクセルのチャンネル値がある閾値を超えた場合は、画像の各ピクセルを白いピクセルに、超えなかった場合は、黒いピクセルに置き換えます。
- 画像を2値画像、つまりシングルチャンネル画像に変換する処理で、画像セグメンテーションの一番簡単な方法です。
- グレースケール化はシングルチャンネル画像の一種です。

画像のフィルタリング処理：畳み込み

- 学習データやテストデータにノイズ除去や画像のぼかしを入れると、汎化性能の高いロバスト（頑健な）なモデル作成や判定を行うことができます。
- openCVでは入力画像とカーネル(フィルタ)のconvolution（畳み込み）を計算できます。以下に5x5サイズの平均値フィルタに使うカーネルを示します。

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

出展：http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering

画像のフィルタリング処理：ガウシアンフィルタ

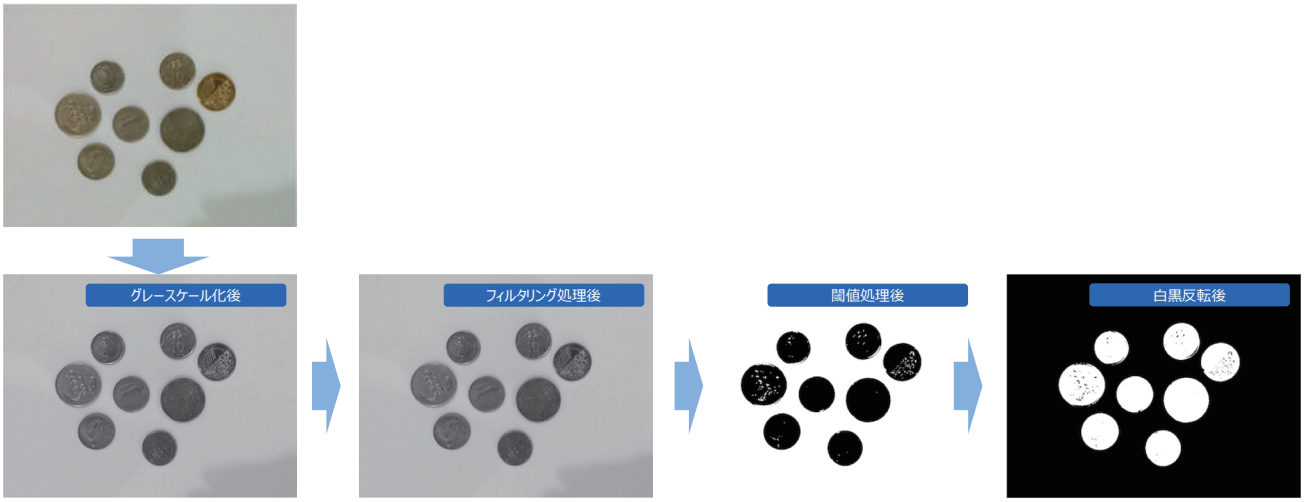
ガウシアンフィルタ（出展：http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering）

- 注目画素との距離に応じて重みを変えるガウシアンカーネルをフィルタに使用する事もできます。
- カーネルの縦幅と横幅(どちらも奇数)に加え、ガウシアンの標準偏差値sigmaX(横方向)とsigmaY(縦方向)を指定する必要があります。sigmaXしか指定されなければ、sigmaYはsigmaXと同じだとみなされます。どちらの値も0にした場合、カーネルのサイズから自動的に計算されます。
- ガウシアンフィルタは白色雑音の除去に適しています。

openCVで実施できる物体抽出、輪郭抽出、形状補正処理

物体抽出

- フィルタリング処理や閾値処理を組み合わせることで、画像中の物体の形状を取得することができます。

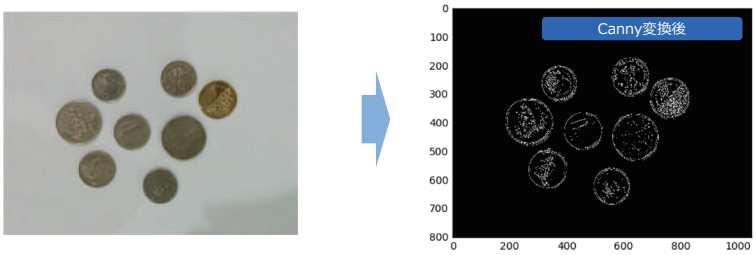


演習1：物体抽出

- ガウシアンフィルタと閾値処理を実施して物体の形状を取り出してください。

境界線の抽出

- フィルタリング処理や閾値処理を組み合わせることで、画像中の物体と背景の境界線を取得することができます。



演習2：境界線の抽出

- Canny変換を実施し境界線を抽出してください。

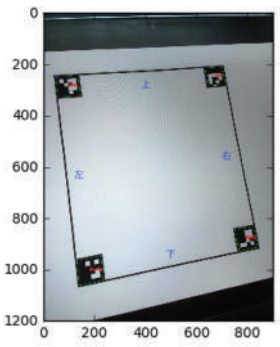
マーカーの認識

- openCVはARマーカーを生成する機能があります。
- 生成したARマーカーには形状ごとにIDが振られており、ARマーカー認識時にIDがも識別することができます。

ID : 0　ID : 1　ID : 2　、、、、、　ID : 49

演習3：マーカーの認識

- マーカーを認識させる処理を実装してください。

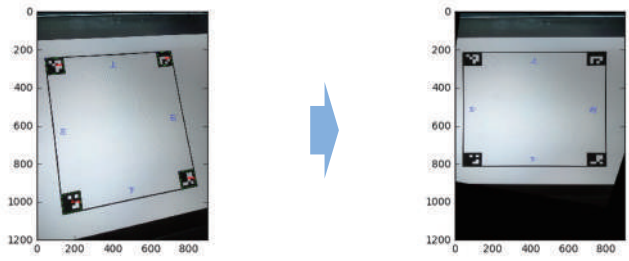


4つのマーカーを認識させた例

マーカーの利用例

- スマホで写真を撮影した場合、被写体の形状が回転、台形変形したり、また光の当たり具合にムラがあったりと、画像認識の精度を低下させるノイズが入ってしまいます。
- 被写体の形状を補正するには輪郭抽出をし、被写体の座標を取得する必要がありますが、グレースケール化やフィルタリング処理がうまく行かずに輪郭が抽出できない場合が多々あります。

- 被写体の四隅に予めARマーカーを仕込んでおくと、画像前処理を精緻に実施しなくても比較的用意にARマーカーを認識でき、ARマーカーの座標を使って形状補正をすることができます。



被写体が台形になり、かつ回転してしまった画像

台形補正と回転補正を実施した画像

演習4：マーカーの利用例

- 検出したマーカ座標を使用し、射影変換を実施して台形補正・回転補正を実施してください。

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

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

AI プログラミングⅡ教材

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