

2021年度

「ぶんせき講習会」

(発展編)

分析における人工知能(AI)
～AIでの課題を解決にむけて～



主催

日本分析化学近畿支部

近畿分析技術研究懇話会

協賛：化学工業会関西支部、近畿化学協会、日本化学会近畿支部、有機合成化学協会、関西分析研究会、化学とマイクロ・ナノシステム学会

2021年11月26日

オンライン

講習プログラム

1.【講習】分析と AI

AI とはなにか、何ができるかについて説明し、分析化学の現場で AI がどのように活用できるのかについて考えます。AI を簡単にアクセスできるようになると、AI を実装した分析化学は革新的なテクノロジーになることができる-exampleを挙げていきます。

(13:30～14:15 / 45 分) (阪大) 大城 敬人

2. 【講習】機械学習とそれを用いるための計画にむけて

AI を分析化学の現場で実装するためには、目標を定めてそれに沿った計画が必要です。そのために必要な、データを取り扱い方、データを処理するためのアルゴリズムなどをまなび、機械学習を用いた分析手法をまなびます。

(14:30～15:15 / 45 分) (阪大) 小本 祐貴

3. 【演習】Python による AI の演習 —

AI の演習として、Python をもちいた実践を行います。環境構築から、プログラム入門をして、データの可視化や機械学習をもちいたモデル作成、予想などをおこないます。ディープラーニングなどをもちいた学習も行います。

(15:30～17:00 / 90 分) (阪大) 大城 敬人/小本 祐貴

受講者の方々へ

この度は、(公社)日本分析化学会近畿支部、近畿分析技術研究懇話会主催のぶんせき講習会発展編(オンライン)に申し込みいただきありがとうございます。ぶんせき講習会「発展編」は Webex によるオンライン開催となります。以下の内容についてご案内いたします。

1. 事前配布資料について
2. Webex 接続テスト(11/22,24)について
3. 当日の受講(11/26)について

ご案内内容についてご不明な点等ございましたら、担当の者(阪大:大城 toshiro@sanken.osaka-u.ac.jp)が対応させていただきます。

1. 事前配布資料について

ぶんせき講習会発展講座の受講内容についての事前学習が可能なように、講習内容についての資料を用意しております。入手方法は、メールにて添付してお送りしております（この資料のことです）。また、演習及び講義で用いるPythonコードはGithubで配布しております。こちらも事前にダウンロードしてお使いください（詳細は、資料末に記載した「利用するGithubに関する説明」をご覧ください）。

資料の講習内容には、演習問題や解説がふくまれております。事前に確認いただき、当日の講習内容の理解の一助としてお使いください。なお事前配布資料の記載内容の一部が、当日の講習内容と変更されることもございます。ご了承ください。

2. Webex 接続テストについて

参加は、ご自宅、職場など、からのPCによるアクセスとなります。なお、有線接続あるいはWi-Fi接続環境が必要です。スマートフォン・タブレットでの聴講については動作を保証することが困難ですので、事前の接続テストを行いますので、各自でご確認ください。ブラウザとしては、ChromeもしくはSafari/Firefoxなどが利用可能です。

自分のPCでWebexを利用可能かどうかは、ご自分のテスト環境でご確認下さい。また、演習に利用するPythonの環境構築を事前に行っていただき、もし問題がある場合、Webexの接続テストの際にお申しだければ個別に対応可能です。

パソコンから流れる音を拾って時間差で流れることがあるので、イヤフォンを着用するようにしてください。なおWebexについての説明は、受講ガイドの付記に資料を記載いたしますので、ご確認ください。（詳細は、資料末に記載した「利用するWebexに関する説明」をご覧ください）

- テスト日程および時間（予定）

11/22（月）10:00-11:00

11/24(水) 13:00-14:00

申し込みいただいた方には、事前にテスト日時における「**Webex の招待メール**」のリンクをお送りいたします。こちらのリンクよりお入りください。

3. 環境構築について

演習には、事前に Python の環境構築が必要です。環境構築は、演習前に行ってください。

環境構築の手順は以下の通りです。

- ①Anaconda をインストールします。(詳細は付記に記載しています)
- ②`https://github.com/ohshirotakahito/bunseki_koshu.git` のリンクから Github で、講習用 Python コードをダウンロードします。(詳細は付記に記載しています)
- ③ダウンロードした zip ファイルを解凍し、解凍フォルダ (bunseki_koshu-main) を例えればデスクトップに配置します。
- ④Anaconda の Spyder を立ち上げ、新規プロジェクト作成を行います。その際、先ほど配置した解凍フォルダを指定します。(詳細は付記に記載しています)
- ⑤演習の*.py ファイルを実行できることを確認してください。

全部できたら環境構築は成功です。もし環境構築がうまくいっていない場合、担当者大城 (ohshirotakahito@gmail.com) に申し出いただかなければと思います。

4. 当日受講について

申し込みいただいた方には、事前に Webex の招待メールをお送りいたします。メールに記載のミーティングに参加するのボタンを押して入室をしてください。

入室後、受講確認のため、入室時のログイン名を「**申し込み登録者名**」でお願いいたします。難しいようでしたら、チャット機能での現ログイン名と登録者名、所属機関について主催者(大城)あてに打ち込みください。

Webex の接続・入室は、13:00より可能です。「ミーティング参加する」のボタンを押して入室してください。Webex の会議時間は、13:00-17:30まで設定しております。講習開始前の9:45より、主催側からの本講習に関する説明が行われます。

<受講中の注意事項> 会議中は、音声はミュートでお願いいたします。

<受講中の質問について>

講師が質問を受け付ける時間を設けます。その時に、举手をしていただければ、講師や進行係の方からお名前をお呼びします。举手は Webex のメンバーの欄から、举手を選ぶことができます。困難な場合、チャットで質問がある旨書いていただか、ミュートを解除していただいて直接よんでいただかなければと思います。

<演習の対応について>

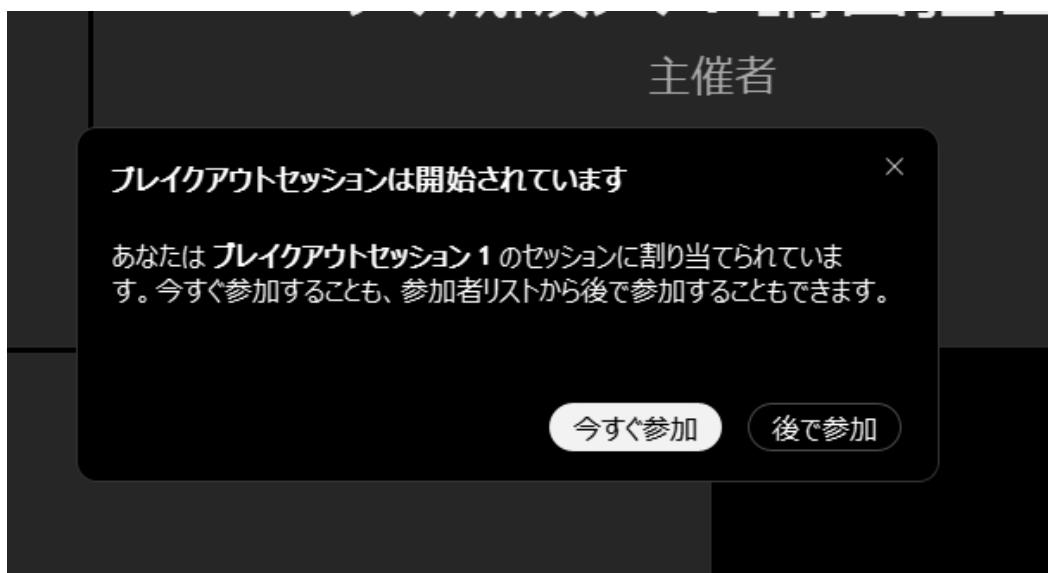
演習は、Webex での講師から演習の司会進行をしています。

Python の演習では、トラブルはつきものですので、遠慮なくして対応を希望していただければと思います。

演習中のトラブルが発生したとき、チャット(図の右下)で「送信先:」を「全員」にして、「**対応希望**」と打ち込んでください。受付係が、メッセージを確認次第、「全員」に「**受け付けました[＊]番**」と、順番に対応いたします。反応が遅い場合、音声で呼びかけていただければと思います。

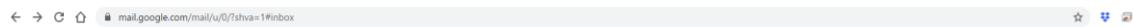


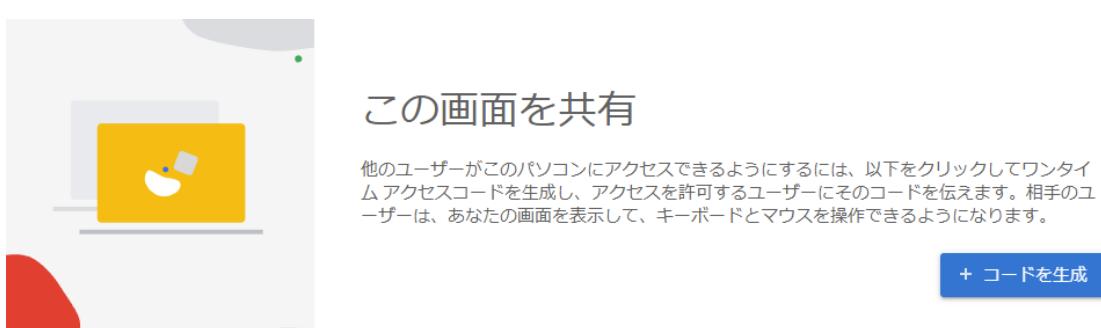
「**Webex のブレイクセッション**」に、こちらのほうから案内いたします。



トラブル解決が口頭では難しい場合、「Webex」のリモート制御や「**Google リモートデスクトップ**」等をもじいて、端末を操作しながら、解決を行います。

もしもの場合に備えて、可能であれば Google リモートデスクトップを導入をお勧めいたします。Chrome のブラウザにリモートデスクトップを導入していただくと、アドレスバーの右に 

 リードデスクトップのアイコンが現れます。押すと以下の画面が現れます。



「コードを生成」を押すと以下ののような 12 行の数が現れます。



これをブレイクラームのチャットに張り付けてください。しばらくすると「制御を許可しますか？」とのウインドウが出てきますので、「許可」をして、お待ちください。



第1部 分析とAI

—AIの概要とPython—

大阪大学 産業科学研究所 大城敬人

e-mail: toshiro@sanken.osaka-u.ac.jp

第1部の内容です.

人工知能と環境構築 —AIの概要とPython—

緒言：機械学習の注目度

機械学習の人気度の推移

機械学習は年々注目を集めています

機械学習の一例

- alpha Go
- 画像の変換
- 手書き数字の認識

Silver, David, et al. nature 550,7676 (2017): 354-359.
Zhu, Jun-Yan, et al. Proceedings of the IEEE international conference on computer vision, 2017.

機械学習は近年注目を浴びておられます。特に、Google DeepMind によって開発されたコンピュータ囲碁プログラム'AlphaGo'が2016年に当時の世界トップ棋士を破ったことが世界的なAIブームを生んだきっかけとなりました。現在AI開発の最前線では、人間の識別限界を超えるアルゴリズムが次々と開発をされています。たとえば、1000万枚の画像の中から、画像識別をおこなってそのエラー率からAIの性能を評価するコンテストでは人間のエラー率4%を2015年に初めて超えました。こうした機械学習の成果や手書き入力文字・数字、言語処理など多岐にわたる分野で応用されその機能は飛躍的に向上しています。

本講習の趣旨

分析化学においても年々機械学習の注目は増加傾向

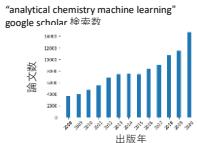
本講習の目的

今後、機械学習を用いた解析を可能するために、初学者が機械学習を知る第一歩として必要な最低限の知識を学ぶ

機械学習とは何か、どんなことができるのかを知る。

機械学習を実際に動かしてみる。

本講習では初学者が機械学習を分析化学に対して使用できるようになることに主眼に置き、精度向上のノウハウや詳細な基礎原理などを学ばない



本講習の趣旨は、分析化学においても年々機械学習への注目が増加傾向である現状を踏まえて、今後機械学習を用いた解析を分析の現場で可能とするために、機械学習の初学者の方がその第一歩としての基礎的な知識を学ぶことです。そこで、まずは機械学習とは何であるかを座学で概要を知っていただき、次に演習を通じて機械学習を実際に行ってもらうこととしました。分析化学では解析の精度向上は必要な命題であることは承知していますが、今回の主題はあくまでも「機械学習で解析できる」ことをめざすものです。

機械学習とは 用語の解説

機械学習に関する用語でAI・機械学習・ディープラーニングなどがある。どのような関係か

AI、人工知能

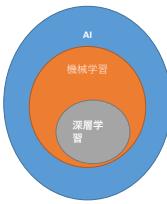
- 機械により知能を実現する理論や技術の研究
機械の推論機能や機械学習、ソフトウェアロボットの研究がある。

機械学習

- データから学習することで、適切な知的ふるまいを人工的に実現すること

ディープラーニング・深層学習

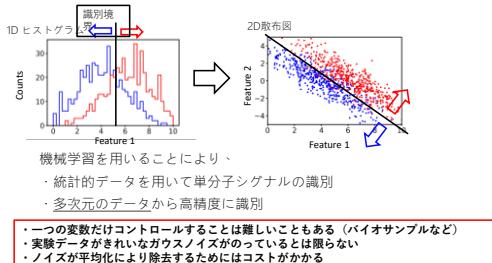
- 機械学習の手法の一つ
多層のニューラルネットワーク



まず、機械学習で用いる用語、特に混乱されている方が多いAI、機械学習、ディープラーニングの用語の整理をしていきます。AIとは機械により知能を有する理論や技術の研究全般を包括したものです。その一分野に機械学習が存在します。機械学習はデータから学習することで適切な知的振る舞いを人工的に実現するもので、本講習はまさにここになります。最近注目されているディープラーニングまたは深層学習はこの機械学習の一つの分野となります。ディープラーニングでは、例えば画像などの入力データ、から猫や犬、馬などの出力データの間に複数の判断する層(隠れ層)の点と点をつなぎだネットワークをもちことで、高度なデータ処理を行うことができるようになっております。

分析化学の分野で、機械学習で用いることが注目されているのでしょうか？それは、分析化学ではセンサーを用い、対象の特徴から分類や定量を行っていますが、1つのセンサーから得られた値、機械学習でいう1つの特徴量からは判断が難しい場合があります。また、センサーから得られた値は多くの場合ノイズが真値とのずれを生じさせ、誤差を生じさせます。このノイズの除去、すなわち精度や確度の向上は分析機器メーカーの方の努力でなされていますが、計測時の状況や測定者やセンサーの経年変化などによってかならずしも、このノイズ除去を行うことが簡単ではありません。これらの課題を、機械学習に克服できないかと注目されています。

なぜ機械学習か



分析化学におけるAI

分析化学におけるAI

- 特徴を抽出 データの次元削減**
用いられるアルゴリズム
主成分分析 (PCA)
独立成分分析 (ICA)
自己符号化器 (Autoencoder)
- 与えられたデータをグループ分けす**
用いられるアルゴリズム
k-means
混合ガウスモデル (GMM)
DBSCANなど
- データを分類**
用いられるアルゴリズム
k最近傍法
Random Forest
深層学習(CNNなど)
- 数値データを予測**
用いられるアルゴリズム
線形回帰
Random Forest
深層学習(CNNなど)

分析化学の分野での課題を大別すると以下の4つになります。一つ目は、データの数が多くなから、必要な情報を抽出する次元削減です。PCAなど分析化学でもおなじみのものです。二つ目が、とってきたスペクトルなどのデータから分子の種類を判断するなどの分類問題で、機械学習では教師あり学習の「分類: Classification」であつかっています。三つ目が、与えられたデータからグループ分けするもので、機械学習では教師なし学習の「クラスタリング: Clustering」であつかっています。四つ目が、与えられたデータから、データの傾向を予測することで数値予想するもので、機械学習では教師あり学習の「回帰: Regression」で扱っております。

分析化学におけるAI - 次元削減

分析化学におけるAI - 次元削減

主成分分析

もし左のデータを1次元にするとしたらどのように基底をとるか？
分散が最大になるように軸をとる
x,yの情報を保持したまま1次元に削減できる

主成分分析を利用したデノイズ

NMR array spectra ($\times 40$)
PCA
Yasunari Kusaka et al. J. Phys. Chem. A 2019, 123, 47, 10333-10338

分析化学の分野での実際に扱っている例を紹介しながら、先の課題内容を確認していきます。まずは次元削減の例でみていきます。例えば2つの独立のセンサーから得られた値(x,y)があったとして、その値をプロットすると図のように図示できます。二つの値はデータで見ると強い相関があることがわかります。このことからデータの傾向は、極端な話 x と y の一方だけでもわかります。この x と y データの情報を保持したまま1次元に削減できるようにするのが主成分解析で行っている。この主成分解析も機械学習の一つといえます。こうした主成分分析を利用したノイズ削減は NMR のノイズデータの削減でも行っており、最近でも論文で報告されています。

分析化学におけるAI - クラスタリング

分析化学におけるAI - クラスタリング

k-means

ランダムにクラスターを分割
各クラスターの重心を求める
最もクラスターの重心に近いクラスターに変更
繰り返し、点が変わらなくなったら終了

教師なし学習を用いた画像、スペクトルデータの分類

スペクトル情報に基づいて、粒子をクラスタリング

J. Ofner et al., Anal. Chem. 2015, 87, 18, 9413-9420

次にクラスタリングを例にとってみていきます。クラスタリングは、教師なし学習という入力データのみからグループ分けをするもので、分析化学ではたくさんのあるIR,NMRなどスペクトルから、傾向の異なるグループに分けていくことがおおくあります。そこで用いられる考え方として、最近傍法というものがあります。最近傍法は新しいデータに対し、最も近い(数個の)既存データが属するクラスターに分類するもので、これを一般化したk近傍法です。これを用いて、合成した粒子や生体分子の画像データをその形や大きさなどの特徴量をもとに空間配置し、距離が最も近いクラスに割り当てています。これによりグループの境界を決めることができます。

具体例(回帰)

回
帰・入力 x が与えられた時の出力 y を推測する

最も単純な回帰問題：線形単回帰 としんせき講習 基礎編

最小二乗法
最小の図中の黒矢印の二乗の和（最小二乗誤差）になるパラメータ w_1, w_0 を求める。

高度な手法(ディープラーニングなどの回帰)
最小二乗法と類似

・損失関数(二乗誤差など)を最小にするパラメータを見つける。

分子情報からLCのretention timeを予測

Liquid Chromatography (LC retention time)
Machine Learning (D algorithm)
Evaluation of t_R predictions

R. Bouwmeester, Anal. Chem. 2019, 91, 5, 3694-370

次に回帰についてみていきます。分析化学でよく用いられる典型的な予想問題です。実際に、実験条件(説明変数)と計測データ(目的変数)のデータプロットを行い、最小二乗法による回帰直線を引くことで実験を行わない条件であっても計測結果を事前に予想できます。この最小二乗法は、回帰直線の最適化の際に用いる手法で、計測結果 y と回帰直線から得られる $y^* = w_1x + w_0$ の二乗したものの、すべての計測結果の和(すなわち最小二乗誤差)を最小にできる w_1 と w_0 を求めます。ある最適な回帰関数に対して、誤差を最小化することは、ディープラーニングでも同様に行っています。実際の分析では液体クロマトグラフィーの溶出時間や順番等を予想できます。

回帰

回帰に用いるデータのイメージ

説明変数 x

目的変数 y

回帰結果の見方
論文で結果を報告する場合、テストデータの値を予測し、真の値と比較される

学習用

テスト用

機械学習

予測結果

Predicted Value vs True value

次に回帰を例にして、機械学習でやっている内容について具体的に説明します。回帰でおこなう機械学習では、入力するデータは、説明変数と目的変数の二つがあります。原因側のデータを説明変数といい、結果側のデータを目的変数といいます。このデータを、教師あり学習では、学習に用いるデータ(上半分)と予測した回帰の結果を評価するテストデータ(下半分)の2群に分けます。この2群の分け方はランダムです。これは回帰予測がデータのもつ偏りに依存して実態と合わないものにならないように評価するため、回帰に用いないデータで回帰の結果を評価する必要があるからです。評価は、回帰との誤差が小さい時良いものといえます。

具体例(分類)

分類
青のデータ●と赤のデータ●を学習 x_1
白のデータ○は青か赤か？

例 画像の認識
手書き数字の識別
←は3???

簡単な分類手法：k最近傍法

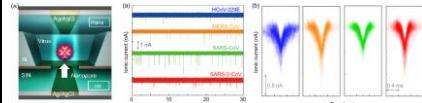
データ間の距離を計算
最も近い(例えば)3個のデータのラベルの多数決
近いものでは赤の方が多い → 赤と決定

分類の仕方は様々であるが、決まった計算の手続きによって分類する。

回帰に続き、分析で多い問題として分類があります。例えばこの図では手書き文字を例に考えていきます。手書き文字で「3」「7」のいずれかに属する画像データを用意して、あらかじめわかっている「3(赤)」と「7(青)」のデータを説明変数ごとプロットしていくことを考えます。実際には多次元です。図のように二つのグループに分かれていってその境界線を書くのが、分類です。一度境界線をかけば、境界線を書くことに使わなかったデータ(○)をプロットしたとき、どちらのグループに属するか判断することができます。この境界線を書く手法が様々あり、最も簡単な方法が、クラスタリングでも用いたk最近傍法になります。これ以外にも様々あります。

分析化学におけるAI - 分類

新型コロナウイルスの計測データ



M. Taniguchi et al.,
Nat. Commun.,
12, 3726(2021)



新型コロナウイルスです。

つづいて分析でこれをもちいた最新の分類例を紹介します。ここで取り上げる論文では、新型コロナウイルスを、ナノボアというウイルスサイズの大きさを持つデバイスを、ウイルスが一つ一つ通過するときに得られる封鎖電流の変化の様子を計測しています。各データ波形をプロットしたものが、青、オレンジ、緑、赤色で示しています。ナノボア計測はコールターカウンターの原理と同じく、得られる封鎖電流の最大大きさはナノボア中の粒子の体積を表すため、サイズの違いの少ないウイルスを識別することは困難でした。しかし、データ波形の違いがあることを、機械学習ではとらえることができるから、識別に成功しています。

分類

分類に用いるデータのイメージ

計測データ		正解分子
X	Y	
2.000000	length?	1.000000
2.000000	length?	1.000000
3.001980	-0.84711	1.377177
3.001980	-0.84711	2.648477
4.001060	-0.112770	0.946144
4.001060	-0.112770	1.187689
5.002030	-0.000000	0.000000
5.002030	-0.000000	0.000000
6.003000	0.842933	1.180480
6.003000	0.842933	3.745204
7.003980	-1.004043	2.018083
7.003980	-1.004043	3.242714
8.004950	-0.737050	1.886402
8.004950	-0.737050	2.548979
9.005920	-0.000000	0.000000
9.005920	-0.000000	0.000000
10.005910	-1.98702	2.008049
10.005910	-1.98702	3.521218
11.005853	1.288863	4.080144
11.005853	1.288863	4.574073
12.007042	1.888138	3.930667
12.007042	1.888138	4.530734
13.007042	2.344035	1.177889
13.007042	2.344035	2.488566
14.007140	2.344035	1.340441
14.007140	2.344035	2.367187
15.007153	4.206329	1.340441
15.007153	4.206329	6.011254

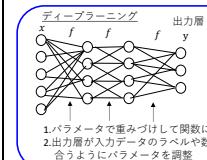
分類結果の見方

論文で結果を報告する場合、主に、検証用に混同行列(confusion matrix)が使用される

		SARS-2						
		True	False	SARS-2	Not SARS-2			
Actual	SARS-2	444	101	52	43			
		96	410	54	80			
		38	85	450	67			
		55	107	90	363			
		SARS-2を分類した時の結果						
		SARS-2と分類された結果の数						

M. Taniguchi et al.,
Nat. Commun.,
12, 3726(2021)

ディープラーニング



fは活性化関数と呼ばれる関数

シグモイド

ReLU

機械学習の最後にディープラーニングについて説明いたします。ディープラーニングの基礎は図で示すニューラルネットであらわされています。左側列の○が入力データを表しており、右側の列の○が出力で、それぞれ分類でいう計測データと、分類名に相当します。左の次元を右側の次元にまで削減する途中に中間層の値を○で示します。この○は、左の全要素から重みwを線形結合した値をfで書かれた関数に代入した値となります。この関数fとして、非線形の関数ReLUなどが用いられます。出力結果の誤差が小さくなるようにwの値を調整する計算を行っているのがディープラーニングです。線で描かれた数だけ重みwが存在するため、膨大の値を調整する必要がありますが、計算機(PC)の進化により実現可能となりました。

Python

Python
Pythonはプログラミング言語のひとつ
機械学習分野に頻繁に使用される。

Pythonの利点
・環境構築が簡単
・ライブラリが豊富
・ユーザーが多い
・フリー

Pythonの欠点
・実行速度が速くない
C言語などで書くのが適した場合も

本講習ではPythonを使用
ただし、使い慣れたソフト(Matlab, Rなど)があれば、各自のやりやすい方法で
やるのを望ましい。

本日の講習の演習では、Pythonをもちいて、機械学習を行います。Pythonはプログラム言語のひとつで、機械学習の分野でも多く利用されています。Pythonの利点としては、無料でプログラムの環境構築が簡単であることが主な理由です。それに伴い機械学習を行うユーザーの数が多く、さまざまなアルゴリズムのライブラリを取り込むだけで簡単に使うことができ、エラーが起こってもその対処法がWeb検索するだけで多くは解決できます。もちろん、他の言語であるMatlabやRなどでも同等の機械学習を行うことも可能ですが、初学者の場合YoutubeなどでPython関連の豊富なコンテンツがあるので、そのメリットを享受するといいと思います。

Pythonの環境構築

Pythonは個別でもインストールできるが、
科学で使うライブラリがひとまとめになったパッケージをインストールすると便利

anaconda
科学計算（データサイエンス、機械学習アプリケーション、大規模データ処理、予測分析など）のためのPythonの無料ディストリビューション
<https://www.anaconda.com/products/individual>

パッケージのインストール
パッケージを追加したい場合、ウェブサイトでインストール方法を確認する。
pipによるインストールというのがあれば
インストールされるAnaconda Promptで
pip install XXX(パッケージ名)
と入力すればインストールできる。



本日の演習で用いるPythonの構築は、機械学習で用いるライブラリを多く含んだAnacondaパッケージをインストールすることで行うこと推奨します。データ読み書き、データ処理、エクセルなどの表計算、図の作成、データの可視化などにはさまざまライブラリが最低でも必要ですが、そのそれもがAnacondaでは含まれています。これにふくまれていないライブラリは、pip install(あるいはconda install)でインストールできます。ライブラリのほかに、コードを書き実行するエディタとして、Spyderが含まれています。演習ではこれを用います。演習で用いるコードはGithubからダウンロードし、対象フォルダをプロジェクトフォルダにしておいてください。以上

第2部 Pythonで機械学習

—Pythonをもちいた機械学習—

大阪大学 産業科学研究所 AIセンター 小本祐貴

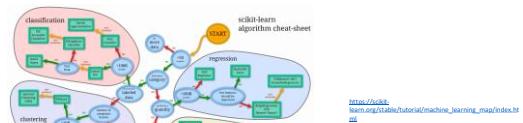
e-mail: komoto@sanken.osaka-u.ac.jp

第二部の内容です。

Pythonで機械学習 —Pythonを用いた機械学習—

機械学習を用いた解析の一連のプロセス

1. 問題を設定する
2. データを取得する。
3. 適切な機械学習の手法を選択する。
実験科学者は以下のようなチャートを参考にして手法を決定すればよい
4. 学習・評価



本講習では、最終的には機械学習を分析化学の現場で実装するために
は、目標を定めてそれに沿った計画を自分で道筋がたてられるようになる
ことを目標としています。そこで機械学習を用いた解析の一連のプロセス
を一つづつ確認していきます。まずは問題を設定します。行いたい解析が
分類なのか回帰なのか目的を決めることです。そしてデータを取得しま
す。これはセンサーから出されるデータと、そのデータの形式をそのまま欠損
などの調整を含むことを含みます。そして用いる機械学習の手法を選択し
ます。はじめての方は図のチャートを参考にするといいでしよう。そして学
習をこない、学習器を作成します。その後作成した学習器を評価します。

目的の設定

問題：ある集団がある疾患に罹患しているかどうか検査する。ある集団は事前に30%が罹患していると推測される。次の3つの手法のうち最もよい手法はどれか。

- 手法A：陽性を50%，陰性を100%で正しく判定
手法B：陽性を90%，陰性を90%で正しく判定
手法C：陽性を100%，陰性を50%で正しく判定

手法A		手法B		手法C	
陽性	陰性	陽性	陰性	陽性	陰性
予測 真実	15 15	27 3	0 63	30 0	35 35
精度	85%	90%	65%	90%	50%

➡ 精度が最も高い手法Bが最もよい手法？

でははじめの目的の設定です。そこで問題です。問題文を読んで、以下に3つの手法のうちどの手法がこの問題に対して最も良い手法と言えるでしょうか？この問題は、ある疾患の陽性か陰性かの分類を行うものです。図は混同行列を示しています。この三つの手法を比べて、精度の最も良い手法は、対角行列が最大となっているのが手法Bです。それでは、手法Bが常に良い手法と言えるでしょうか？というのが問題です。

良い機械学習とは何か

手法A		手法B		手法C	
陽性	陰性	陽性	陰性	陽性	陰性
予測 真実	15 15	27 3	0 63	30 0	35 35
精度	85%	90%	65%	90%	50%

✖ 精度が最も高い手法Bが最もよい手法
○ 最もよい手法は目的によって変わる。

- ケース1) 確実に陽性患者を見つけたいとき → 手法A
ケース2) 感染者数を推測したいとき → 手法B
ケース3) 疑いのある者は全員隔離したいとき → 手法C

統計・機械学習は目的を設定してくれない。
機械学習の目的は人間が決める

先ほどの答えです。最適な手法は「目的によりかわる」です。精度の高さに目がいきがちですが、次のケース3つの目的について考えてみてください。ケース1ではなによりも偽陽性をさけることが優先されるため、手法Aが「よい」手法です。ケース3では偽陰性をさけることが優先されるケースで、手法Cが「よい」手法です。例えば致死性ウイルスに対する防疫を目的として、患者を絶対みつけることを優先することに場合に相当します。擬陽性・偽陰性をいずれも低く抑えたい時には手法Bが「よい」手法と言えます。したがって、機械学習の手法の選択についても目的を設定しておくことが「良い」機械学習を行う上で重要となります。

データの収集

サンプルサイズはどれぐらい必要か

→データによる

機械学習は多くのデータがある方が精度が良くなるが、データの質がよければ少なくともよい

→目的による

精度が99%必要か、80%必要かは目的に依存

先行研究のサンプルサイズ

太陽電池の効率を予測

特微量 13次元

実験系 280 手法 Deep Learningなど

H. Saito et al. Adv. Energy Mater. 8, 180202 (2018)

单分子計測によるDNA塩基の識別

4塩基 特微量 10次元

シグナル数 1000 手法 Random forest

M. Taniguchi et al. Phys. Chem. C 2019, 123, 15867–15873

目的が設定した後、データの収集に入ります。まずデータサイズはどのくらい必要なのでしょうか？データは学習器を作成するうえで重要なものです。一般的にはデータ量は多いほうが、よい学習器ができるますが、データの質が高い場合、必要なデータ量はそれほど多くなくてもできます。また、目的によってはデータ量が変わります。これは出来上がる学習器の必要な精度が目的によって変わることからです。これ以外にも、選択する機械学習の手法によっても必要なデータ数が変わってくる。ディープラーニングは高い精度を出すために多くのデータが必要だといわれています。

データの前処理

- データを機械学習に用いることができるようになります。
- ・欠損データの除去
 - ・実験データの修正 ex.各種較正
ラマンスペクトルの宇宙線ピークの除去など

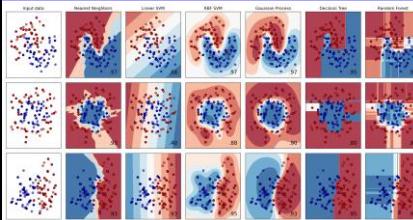
データの標準化

- ・各データの平均を0,標準偏差が1になるようにデータを変換



データを収集したら、次に行うのはデータの前処理です。分析の現場でもセンサーラ装置が途中で壊れるなどやもえない事情によりデータの欠損が発生することがあります。機械学習では、欠損データをどのように処理するのかによって精度は大きく変わってきますので注意する必要があります。欠損値を0とするか、平均値をいれるか、削除するか、などさまざまな方法があります。また、データの標準化もデータ前処理の重要な要素となります。各入力データには物理量が異なるため、単位も大きさも異なります。計算機はこうした入力は数値でしか判断できないため、入力特徴量の重要度を計算機に誤解させないためにデータの標準化が重要となります。

アルゴリズムの選定

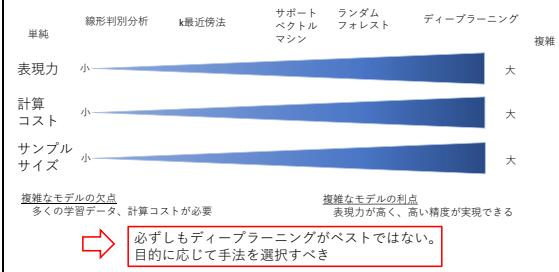


https://scikit-learn.org/stable/auto_examples/cf_classification/plot_classifier_comparison.html

アルゴリズムによっては原理的に識別不能な場合がある

次に行うのが機械学習の手法の選択です。この手法のことをアルゴリズムと言います。この図は、二つのグループ(赤い点と青い点)が、アルゴリズムによってどのように識別境界線(赤と青色の領域の境目)が描かれるかを図示したもので、入力データが同一でも選択したアルゴリズムによってかなり異なる識別能となることが、識別境界の違いで判ります。これは各アルゴリズムの持つ関数の表現力に起因しています。いくつかのアルゴリズムでは、識別境界線の曲線表現が苦手であることがわかります。このようにアルゴリズムによっては原理的には識別が不能な場合があります。

アルゴリズムの選定



それでは、アルゴリズムの表現力があるものを常に選択すべきなのでしょうか？次の表は、各アルゴリズムの表現力と計算コスト(要する計算時間や計算機の能力)とサンプルサイズ(データ数)を示したものです。表現力の点でディープラーニングは優れています。しかし、必要な計算コストや必要なサンプルサイズが大きくなります。特にサンプルサイズは100万を超えるデータが必要なケースが多く、データが100程度しか取れない場合適切なアルゴリズムとはいがたいです。表現力が小さいものであっても十分な精度を出せるケースもあるため、やはり目的に応じて手法を選択すべきだといえます。

学習

機械学習用に前処理されたデータを機械学習に読み込ませる。
機械学習のパッケージを使うのが便利

頻繁に使われるパッケージ
機械学習全般 scikit-learn
深層学習 TensorFlow, keras, Pytorch など

パラメータチューニング
学習で変化せず、前もって決めておくパラメータ(ハイパーパラメータ)がある
グリッドサーチ (考えられるパラメータの総当たり)などをしてよいパラメータを決定

Random Forest, SVM : パラメータが少なく、比較的容易
深層学習 : パラメータのチューニングが難しい

機械学習で用いる多くのアルゴリズムは scikit-learn にすでにパッケージとしてはいっており、本講習ガイドに記載した Anaconda をインストールされたかたは自動的にこの scikit-learn をふくむパッケージに含まれています。ディープラーニングのパッケージも Python でも使用することができ、tensorflow, keras, Pytorch などを、コマンドプロンプトから conda(pip) install (パッケージ名)でインストールしておけば、使えます。機械学習では関数のパラメータを計算機が最適化しますが、そのパラメータの数や関数の種類を決めるハイパーパラメータを人の方が設定する必要があります。Random Forest, SVM ではパラメータが少なく初学者でも容易に扱えます。

モデルの評価

過学習
機械学習では未知のデータが予測できるようにするのが重要
必ずしも最小二乗誤差が小さいからよいモデルとは言えない
訓練データに特化しすぎ(過学習)ないようにする。

k-fold 交差検証(Cross Validation)
データをk分割する
1つをテスト用とし、その他を学習
すべてのデータが1回テストされるようにk回学習

1st	train	train	test
2nd		train	
3rd			train

アルゴリズムを選択し、データを入力すると、計算機がパラメータを調整し、学習器が出来上がります。出来上がった学習器はここではモデルといいます。モデルの良し悪しは、「未知のデータ」に対してたどり着いた予測ができるかということになります。そこで次の2つの回帰モデルの図を比較してください。同一のデータですが、赤線で示す予想回帰がずいぶんことなります。最小二乗誤差でいうと左より右の回帰は小さいですが、右の予想回帰モデルはよいのでしょうか？これは、訓練データに過度に特化した回帰モデルで、良いモデルではないといえます。これを「過学習」といいます。これを避けるためテストと学習データをいくつか分割し、すべてのデータで比較する交差検証などにより作成したモデルを評価する必要があります

モデルの評価

分類の性能評価指標
ある手法を用いてある疾患が陽性かを市中でランダムに検査します。この手法で、99%の検査者を正しく判定できました。この手法はよい手法でしょうか？

A.これではよいとは言えない
市中である疾患の陽性率が1%であった場合、全員陰性と判定すれば精度99%

True	Predicted	
	Positive	Positive Negative
Positive	TP	FN
Negative	FP	TN

適合率(precision)
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
 F値:適合率と再現率の調和平均
陽性と予測したものが本当に陽性か

再現率(recall)
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
 両者を評価できる
本当に陽性のサンプルを陽性と予測できたか

ここでモデルの性能評価指標について、例をとて考えてみましょう。ある手法を用いて、ある疾患の陽性・陰性 99%の正解を出したとします。これは、よいモデルでしょうか？これは実はよいモデルではないかもしれません。それは、疾患の人の割合(陽性)が1%で、モデルが常に陰性と判定すれば正解率は 99%となってしまいます。この正解率では正しいモデル評価ができないことがあります。そこで適合率と再現率の二つの指標で見直してみましょう。この指標はそれぞれ擬陽性(FP)を減らしたい場合、偽陰性(FN)を減らしたいときに用いる指標です。両者はトレードオフの関係で、その調和平均となるF値が両者ともに評価できるものといえます。

機械学習プロセスのまとめ

・目的を明確にする。

- ・目的にあった手法を選択する。
簡単なものから実際に色々な手法を試してよいものを探る
- ・適切な評価をする

ここで紹介してきた内容や用語は機械学習ではよく用いられ重要なものです。これらの手法の選択や評価の仕方も含めて重要なのは、分析で機械学習を行う前に、まずはどんなことをしたいのかという目的をまずははつきりさせることです。ここから、手法の選択やそれとともにデータの取得量や評価する際に重要となる指標がきまっています。これが、機械学習を分析化学の現場で実装するためには、目標を定めることが重要であるという意味となります。

第3部 PythonによるAIの演習

—Pythonプログラムの基礎—

大阪大学 産業科学研究所 AIセンター 小本祐貴/大城敬人

e-mail: komoto@sanken.osaka-u.ac.jp

Pythonプログラミングの基本

Pythonで分析を行えるようになるために

第三部の内容です。つづきまして、演習の準備に入ります。

演習では先で紹介したPythonをもちいて行います。

プログラミングの基礎

プログラミング一般
プログラミングは上から下へ1行1行実行される
データの型がある。
例えば数字でもint(整数)型, float(浮動小数)型など
=は代入
数学の = は ==

Python
インタープリター方式
C言語などのコンパイラ方式と異なり、実行するためにexeファイルなどを作成しない
インデント（字下げ）によってブロック構造を指示
型は明示しなくてもよい

Pythonで機械学習を行うためには、コードの作成をすることになります。これをプログラミングといいます。基本的にプログラムは上の行から一行一行実行されます。これを順次実行といいます。実行できない場合エラーが帰ってきて実行が中断されます。このエラーを解消しながらコードを作成します。計算機で扱うデータは、数字、文字列、布尔といわれるデータの型があります。数字の型にもint(整数), float(浮動小数)という型があります。型が異なると実行できないことがあります。Pythonはインターパリター方式の言語で、コードの実行にコンパイル作業は不要です。

Pythonプログラミングの動かし方

1.ソースコードを書いたファイルをXXX.pyという形式で保存

ソースコード:書いたプログラムのこと

メモ帳などのテキストエディタでも書けますが、専用のエディタ（Spyderなど）を使うと便利です。

2.Pythonを実行

Spyderの画面



- 1.ここにソースコードを入力
- 2▶でプログラムを実行
- 3.ここに実行結果が表示

Pythonのソースコードを記述したら、実行環境が必要です。Anacondaでは簡単に実行環境を作ることができます。Anacondaと一緒にインストールされるSpyderをつかると、エディタでコードを書いて、左上の▶を押すだけでソースコードの保存や実行を行なうことができます。実行結果は、右下のコンソール画面と記載されます。計算結果やエラーの表示はここで確認できます。データの可視化でプロット図がある場合、右上のコンソールのプロットのタブを押すと表示されます。

数値

整数(int)、浮動小数(float)、文字列(str)などがある

ソースコード

```
#整形の定義
a = 2
print('a=' + a)
#浮動小数の定義
b = 3.5
print('b=' + b)

#計算
c = a+b
print('c=' + c)

#文字列の定義
print('abc')

#文字列の表示
c = "test"
print(c)

#文字列の足し算
c = "test" + "test"
print(c)
c = "test" * 3
print(c)
```

実行結果

```
a = 2
b = 3.5
c = 5.5
test
testtest
testtesttest
```

それは入力データである整数、浮動小数、文字列について実際にSpyderのエディタの書き込んでいきましょう。ソースコードの#のついていない行を書いてみましょう。行の先頭に#がついている場合その行は実行せずにスキップするもので、コメント等を追記や一時的に実行したくないコードを指定するために使います。文字変数に数字を代入するのに「=」をつかっています。表示には「print」をつかいます。

関数とクラス

関数

何回も同じ処理を記述しなくてよいように、全体の一部をまとめたもの

関数の定義
def func(x): #xに3を足して12掛け、値を返す関数
 t = (x+3)*12
 return t

クラス

関数をまとめたもの*

```
class class1(object):
    def __init__(self):
        self.parameter1 = 1
        self.parameter2 = 2

    def calculate(self,x):
        self.parameter1 = x

    def analyze(self,x):
        return self.calculate(x)*self.parameter2

    def printHello(self):
        print("Hello World")

    def __del__(self):
        print("Object is deleted")
```

*クラスには継承などの機能があり、この説明は不十分だが、機能のライブラリを使うための基礎版として、クラスの中にあるメソッド(関数)を利用するということだけ理解して頂く

次に関数の作成と実行をしてみましょう。関数とは決まった処理を行うもので、同じ処理を複数回実行するときに用いると便利です。例えば、値xを入れることで出力tがなされる関数を作成します。行先頭に「def」を書き関数名(なまえはなんでもよい)の「func()」で、()内には代入する値xをいれて「:」とし改行します。次の行は自動的にインデントされ、そこにxの処理内容、ここでは $(x+3)*12$ とすると、出力となるtに代入する「t = $(x+3)*12$ 」と記載し改行します。そして最後に関数の出力となるtを返すという意味で「return t」と記述して改行して終了です。この関数は同じコード内では「func()」をもちいいれば何度も使用できます。

ライブラリの読み込み

```
import ライブラリを読み込み

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_boston |
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

numpyをnpという名前でインポート
以後 numpyの中のaverageを使いたいときは
np.average(X) とすればよい

from sklearn.datasets import load_boston
でsklearn.datasetsの中になるload_bostonだけがインポートされる
```

次にライブラリの読み込みをしていきましょう。ライブラリは関数をまとめてあるソースコードからなってモジュールで、無料で公開されています。ライブラリが計算機にインストールされていれば「import (ライブラリ名)」で読み出すことができます。ライブラリの中の関数をたとえば、numpyのなかの平均をもとめる関数「average」を使う時、numpyをnpという略称でインポートする「import numpy as np」とすれば、平均を求める関数「average」は「np.average()」とすれば()内に数字の配列xを入れる「np.average(x)」だけですべての平均を求めることができます。

ライブラリの使用

特別な目的がなければ、自分で書かずに、できる限りライブラリを使いましょう。

n=1000000, a = np.ones(n)/n : 1/nがn個連続した配列aの総和を求めるプログラム

自分で書く場合

```
s = 0
for i in range(n):
    s += a[i]
print(s)
```

実行時間: 0.74 sec

ライブラリを使

```
print(np.sum(a))
```

実行時間: 0.0030sec

自分で書かない方が短く、正確で早い

やっていただいたように、ライブラリの使用は簡単で、積極的に使っていくことをお勧めします。ライブラリは、「pip install (ライブラリ名)」で簡単にインストールでき、「import (ライブラリ名)」で簡単に実行できるようになります。また、実行時間も早く、ライブラリを使いこなすことができれば、やってみたい作業のほとんどを自分で関数を作成しなくても実行に早く移すことができます。

数値計算ライブラリ numpy

数値データなどは numpyの ndarrayをつかうとよい
Pythonの標準ライブラリのlist型よりも計算向き

ソースコード

```
#ライブラリのインポート
import numpy as np

#numpyのndarray型の定義
y = np.array([1,4,6,5,7,10,3])
x = np.array([1,1,5,2,3,5,3,5,4])

#ndarrayの計算
print("x+y")
print("x*x")
print("x*x2")
print("x*y",x*y)
print("x*y",x*y)
```

実行結果

```
x+y [ 2.  2.5 3.  4.  4.3 4.5 5. ]
x*x [ 2.  3.  4.  6.  6.6 7.  8. ]
x*x2 [ 2.  5.5 8.  8.  10.3 13.5 7. ]
x*y [ 1.  6.  12.  15.  23.1 35.  12. ]
```

次にPythonの機械学習でつかうライブラリをいくつか紹介します。「numpy」は数値計算を効率的に行うことのできるモジュールでMATLABに近い機能性をもっています。np.array()は配列を扱うことができます。配列同士の計算も行うことができます。実行してみましょう。

テキストデータの読み込み

実験結果はしばしば、テキストデータで保存される。
.txt,.csv,.datファイルのようなメモ帳で開くことができるファイルの読み込みを扱う
numpyのloadtxtを使う
他にもpandasなども使いやすい。
好きなものを使えばよい

```
data = np.loadtxt(file,delimiter=',',skiprows=1)
x = data[:,0] ファイルのパスが記入された文字列
y = data[:,1] データの区切り文字タブなら\nなどで指定 最初の1行目を読まない
```



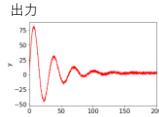
Numpyをもちいて、数値データの読み込みを行ってみましょう。実験結果は、txt.csv, datファイルのような、テキスト形式でのデータが格納されている場合が多いと思います。これらのファイルをnumpyでは、loadtxtで読み込むことができます。やってみましょう。ファイルの読み込みは、np.loadtxt以外にもpandasなどでも行えます。テキストの形式や、デリミタの設定、ヘッダーの読み込みの有無など設定は慣れない方は難しいですが、実験データの形式が分かれれば、一度設定すれば、ファイルpathの設定以外は同じ作業を繰り返すだけですので、頑張ってやっていきましょう。

結果の可視化

読み取ったデータをグラフにする。
結果の可視化にはmatplotlib.pyplotが便利
その他のライブラリ:seabornなど

```
plt.plot(x,y,'r')
plt.xlim(0,200)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

棒グラフ:plt.bar() ヒストグラム:plt.histogram() 散布図:plt.scatter()
など様々なグラフが書けます。



データを読み込み、関数でデータ処理をして出力をするところまで来ました。次は、そのデータを可視化することです。データの数値出力はコンソール画面でprintで表示することができますが、数字の羅列ではわかりにくくともあります。Excelなどのグラフ表示をするモジュールとして、
「matplotlib.pyplot」がよくつかわれます。コードの先頭に「import matplotlib.pyplot as plt」として、xとyの数値または配列を入れてみると
「plt.plot(x,y)」とラベルや大きさを指定した後に「plt.show()」と記述すると、
コンソールのプロット画面に右の図のようなプロット図が表示されます。これ以外に棒グラフやヒストグラム、散布図などが簡単に描画できます。

scikit-learnを用いた機械学習

深層学習以外の機械学習にはsklearnが便利で扱いやすいです。
深層学習は扱えませんが、sklearnに慣れれば、kerasやpytorchなどの深層学習ライブラリも扱いやすい。

```
#機械学習ライブラリの導入
#sklearnのensembleの中にはRandomForestClassifierがあります
from sklearn.ensemble import RandomForestClassifier

#機械学習の特徴量
clf = RandomForestClassifier()#引数指定がない場合はデフォルトのパラメータで指定される

#学習
clf.fit(x_train,y_train) #この中にデータを学習。大分類題で分類ができるようになります。
#引数
y_predict=clf.predict(x_test) #学習した分類題で、testを一つづつ分類してy_predictに入れます
x_train,y_trainを用いて学習
x_testの分類
```

x: 特徴量 次元数×サンプルサイズの2次元配列(行列)
y: ラベル サンプルサイズの配列、各要素はクラスを表す整数や文字列

機械学習でもちいる手法SVM, Random Forest, XGboost, k-meansなどは「sklearn」のモジュールに入っています。例えばRandom Forestを用いたい場合、モジュールを読み出すときは「from sklearn.ensemble import RandomForestClassifier」とし、「clf = RandomForestClassifier()」として、clfにランダムフォレストの関数にします。その後、「clf.fit(説明変数,目的変数)」をいれて学習し「clf.predict(テストデータ)」をいれれば、たったの3行で機械学習ができます。SVMでも同じですし、基本的流れはどれも同じとなります。

書き方がわからない場合は

1. googleなどで使えるものがないか探してみる
2. ライブリヤーがある場合、公式のドキュメントなどを確認



Pythonでやるメリットはとにかくユーザーの数がおおくわからない場合、検索するだけで多くの情報得ることができます。自分でコードをかかなくても、検索したコードをそのまま張り付けるだけでも動くものもあります。はじめはコードの切り貼りをしながら覚えていくとよいと思います。

プログラムが動かない場合は

エラーで動かない場合

エラーメッセージを確認

エラーの例

`SyntaxError` 文法エラー、が掛けている。0が閉じていない、ifの処理が書かれていないなど
`IndentationError` インデント (tab) の位置が間違っているなど
`NameError` 定義されていない変数が出てくる

- ・スペルや型があっているか確認する
- ・ドキュメントに戻って、使い方が正しいか確認する
- ・()の指している場所が正しいか確認する

間違ったプログラムを動かしてもエラーが出るだけなのでトライ＆エラーでやる。

また、プログラムを作成するうえでエラーはつきものです。エラーメッセージを確認し、自分のコードをじっくりと見直すことが必要です。文法の間違い、インデントの間違い、変数名の間違い、関数名の一文字違い、データの型の違いなど一つ一つ確認するだけで多くの場合わかつてきます。またエラーコードをGoogle検索することも有効です。この2つをやるだけで退官としては9割以上解決できます。トライ＆エラーを根気よく続けてみてください。これまでpythonの機械学習をやる前の準備がととのいました。

演習

1. `enshuu_numpy1.py`の出力結果を予測した後に実行してください
2. `enshuu_numpy2.py`を開き、
aを平均が0、標準偏差が1になるように変換してください
3. `koushuu4_load.py`を開き、
 - 1次のプログラムを実行して、`GraphData1.csv`のグラフを確認してください。
 - 2プログラムを一部変更して、グラフを描画してください。
i)x,yは1000点のデータからなります。yの800点以降のデータを0にしてください。
ii)xの値が50以下の範囲のみ青く描画したグラフを追加してください
4. `enshuu_Classification.py`を開き、
RandomForestを用いて`testdata`中にある`Cif2_Train.csv`を学習して、`Cif2_Test.csv`のデータを分類して、F値を評価してください。 (c.f.講習5)
5. `enshuu_clustering.py`を開き、
 - 1次のプログラムを実行してください。
 - 2)KMeans法を用いて`testdata`中にある`Cluster2.csv`を
3クラスターに分けてください。
 - ii)結果を評価してください

これから演習をおこなっていきましょう。

(付記) 利用する Webex に関する説明

Webex

<https://www.webex.com/ja/videoconferencing.html>

アカウントがなくても利用可能ですが、アカウントは「無料で試す」より登録できます(図1)

図1 ログイン・ユーザー登録



図1 ログイン・ユーザー登録

図2

Webex アプリで参加される場合は上アプリで参加される場合は上の赤丸(図2)を、Web ブラウザで参加する場合は下の赤丸部分をクリックしてください。なお、遅延などの問題があるため、Webex アプリでの参加を推奨します。Webex アプリは最新版をご利用ください。すでにアカウントをお持ちで Webex アプリを常に利用される設定の場合は、自動的に Webex アプリが起動します。

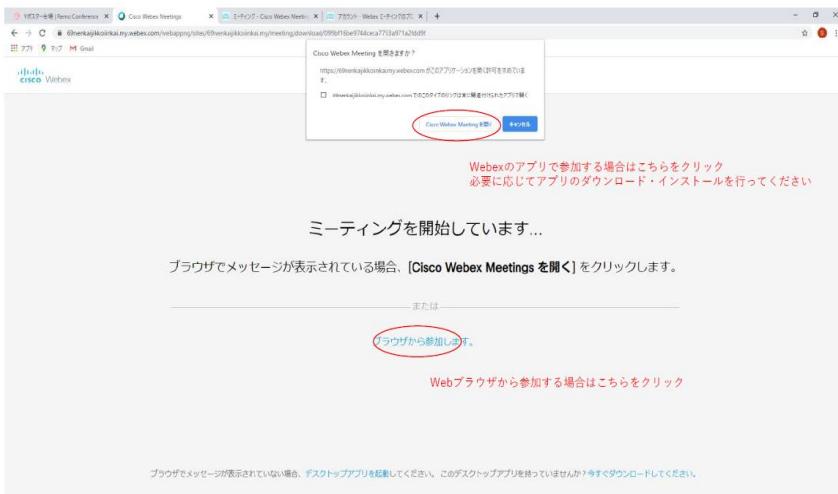


図3



登録メールアドレスに、「Webex ミーティング招待状:分析講習会「その1」」の表題のメールが届きます。「参加する」を選択すると、図3のような画面が現れます(図3はWebexアプリです。ブラウザの場合は類似の画面がブラウザ上に表示されます)。名前とe-mailアドレスを入力してください。名前は「名前:所属」(例:分析太郎:分析大学)としてください。

(付記) 利用する Anaconda に関する説明

演習は windowsPC (Windows 10) をもちいた Anaconda をインストールしたものを用いていくことを前提に Anaconda のインストールの仕方について説明いたします。

① Google で Anaconda を検索.

<https://www.anaconda.com/products/individual> にとぶ

Google 検索結果
検索語: Anaconda

検索オプション: すべて ニュース 動画 画像 ショッピング もっと見る ツール

検索件数: 約 60,900,000 件 (0.49 秒)

検索結果:

- [Individual Edition - Anaconda](https://www.anaconda.com/products/individual)
Anaconda Individual Edition is the world's most popular Python distribution platform with over 25 million users worldwide. You can trust in our long-term ...
Open Source · Enterprise Edition · Anaconda Embedded · Use Cases
- [Anaconda のインストール: Python環境構築ガイド](https://www.python.jp/install/anaconda/)
Anaconda はデータサイエンス向けの環境を提供するプラットフォームです。科学技術計算などを中心とした、多くのモジュールやツールのコンパイル済みバイナリファイル ...
Windows版のインストール · MacOS版のインストール · Linux版のインストール

関連検索:

- 他の人はこちらも検索
- Anaconda インストール anaconda 仮想環境
- AnacondaVSCode anaconda インストールしたら
- anaconda 日本語 anaconda アンインストール

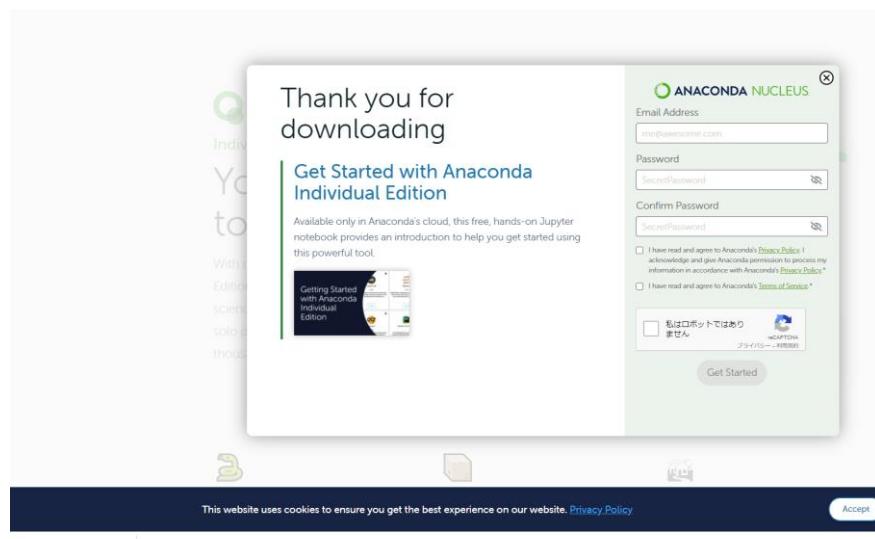
② このページの一番下に行く.



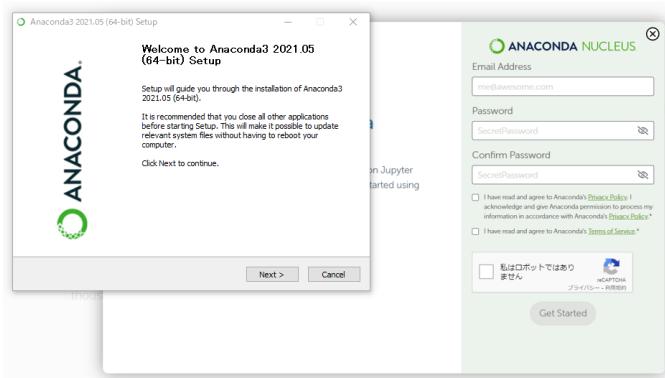
③ このなかで、windows Python 3.9 の installer を選択



④ 左下のように自動的にインストーラーがダウンロードされる。



⑤Anaconda のインストーラーを立ち上げ、インストールを行う。



(付記) 利用する Github に関する説明

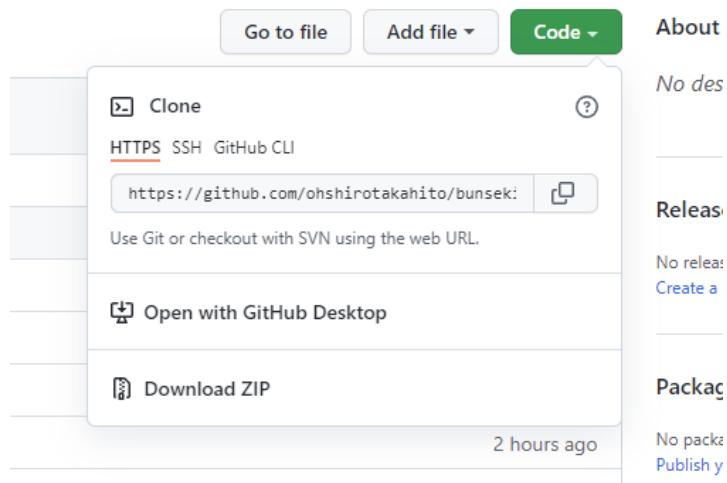
演習用 Python コードは、Github で配布いたします。

①まずは以下のリンクにアクセスしてください。

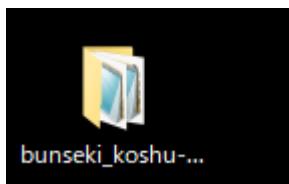
https://github.com/ohshirotakahito/bunseki_koshu.git

The screenshot shows the GitHub repository page for 'ohshirotakahito / bunseki_koshu'. The 'Code' tab is selected, displaying a list of files and their upload history. The list includes various Python scripts such as 'ConfusionMatrix.py', 'Omake_BreastCancer(Keras).py', and 'Omake_seaborn_heatmap.py'. Each file entry shows the date it was added via upload. To the right of the code list, there are sections for 'About', 'Releases', and 'Packages', all of which indicate no activity.

②緑色の Code とかかれた部分をクリックし、プルダウンメニューから、Download ZIP を選択してください。



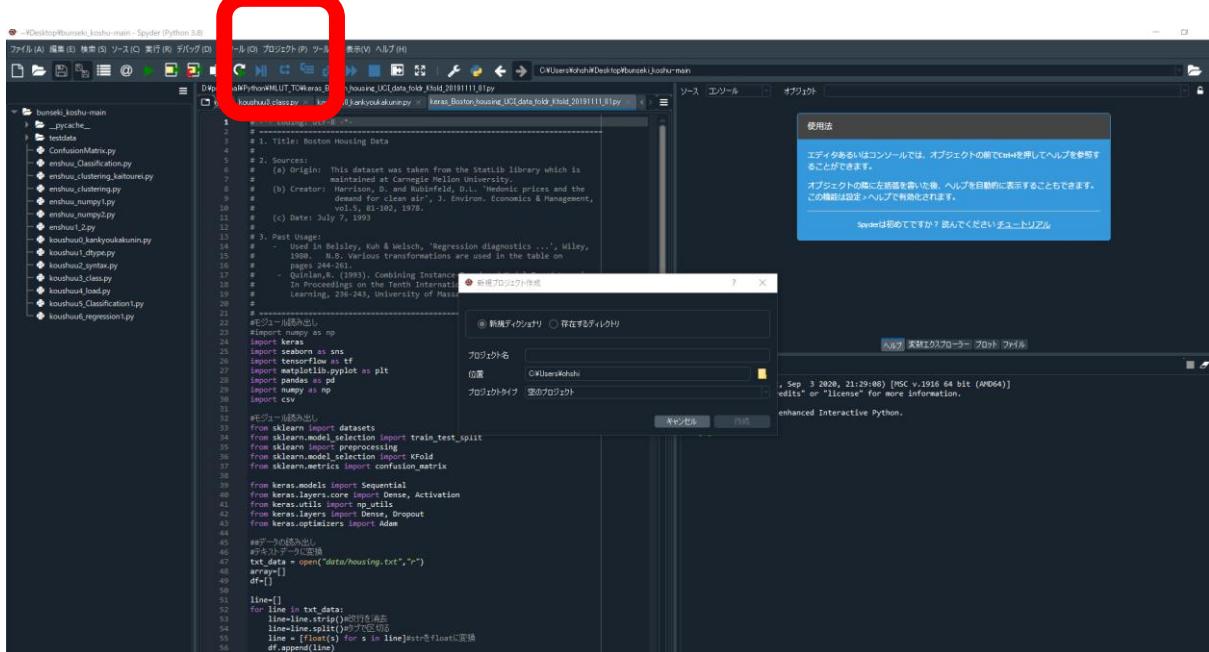
③ダウンロードファイルを解凍します。ファイルの格納場所はどこでもいいです。（例えばデスクトップでもいいです）



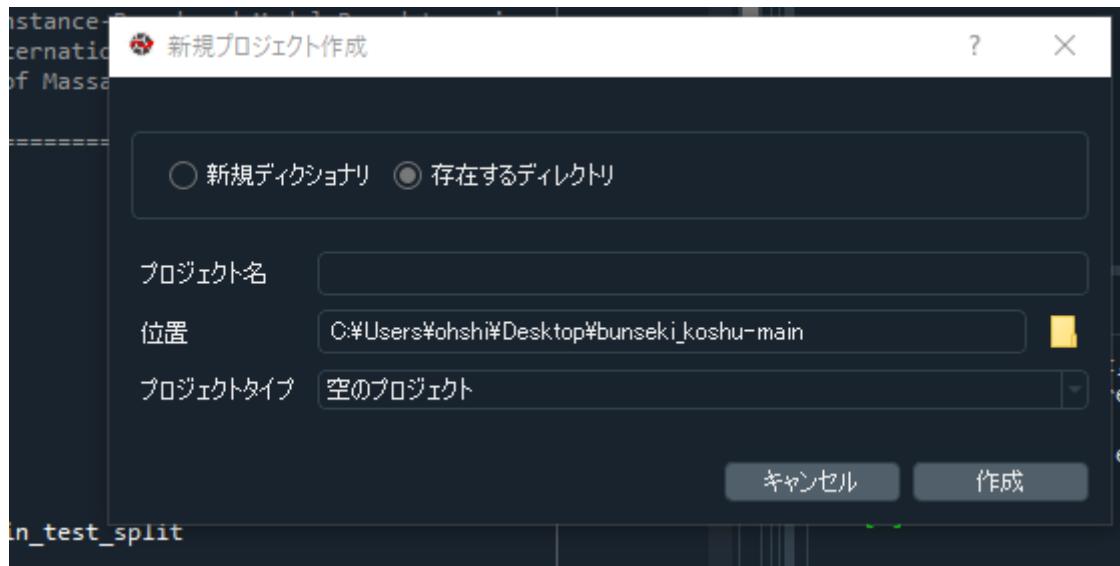
④Anaconda の Spyder を起動してください



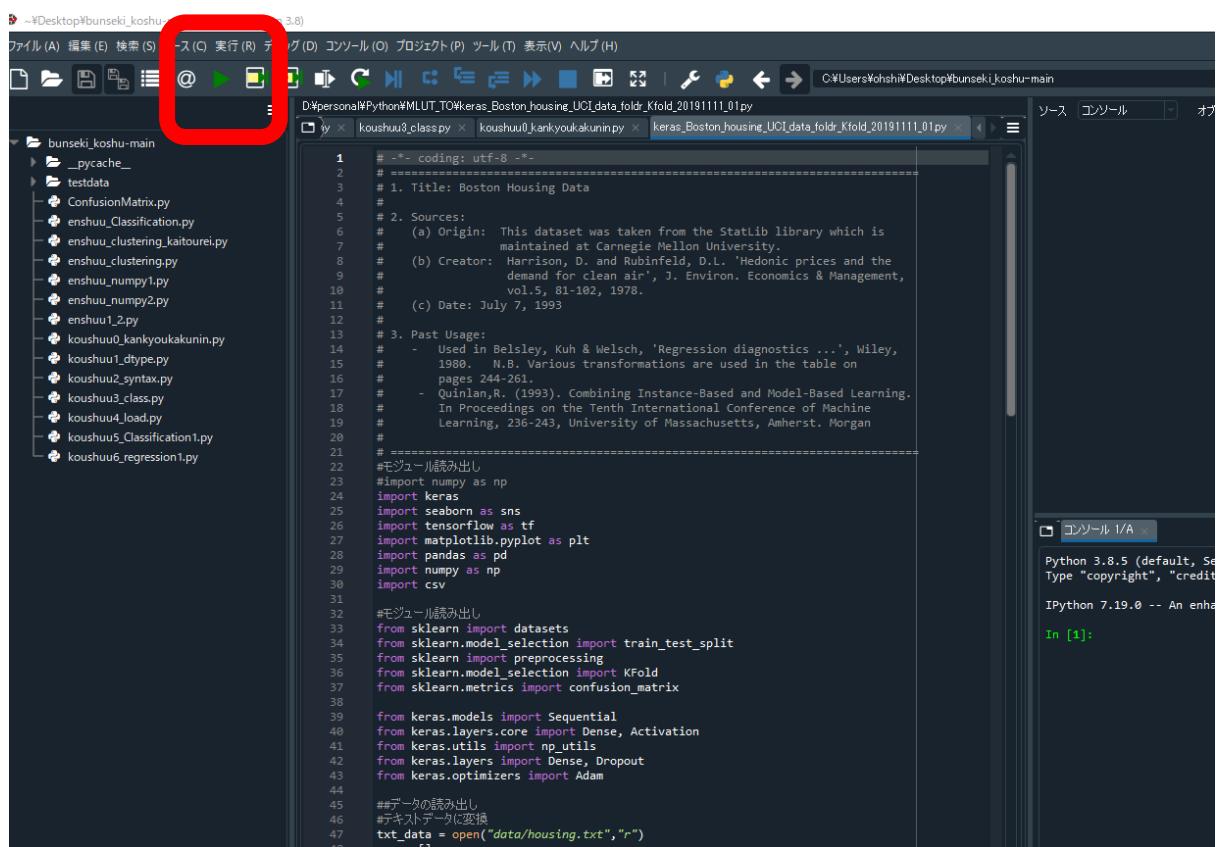
⑤Spyder のプロジェクトから新規プロジェクトを選択します。



⑥新規プロジェクト作成で、存在するディレクトリを選択し、プロジェクトの位置を解凍したフォルダ（bunseki_koshu-main）として、「作成」のボタンを押します。



⑦コードが動くことを確認します。緑の▶ボタンを押してコンソールでエラーがないことを確認してください



(付記) 利用する Chrome リモート デスクトップに関する説明

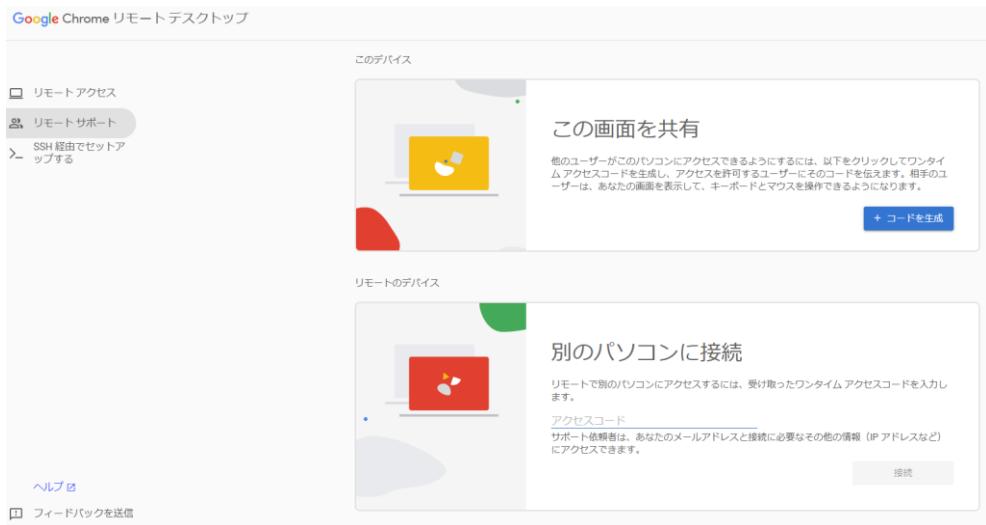
①Chrome のブラウザのリモートデスクトップを事前に使用できるようにしてください。



インストールできたら、次の画面から、リモートでの対応が必要な時にリモートデスクトップを開いてください。



②「この画面を共有」の「コードを生成」で、コードを取得してください。



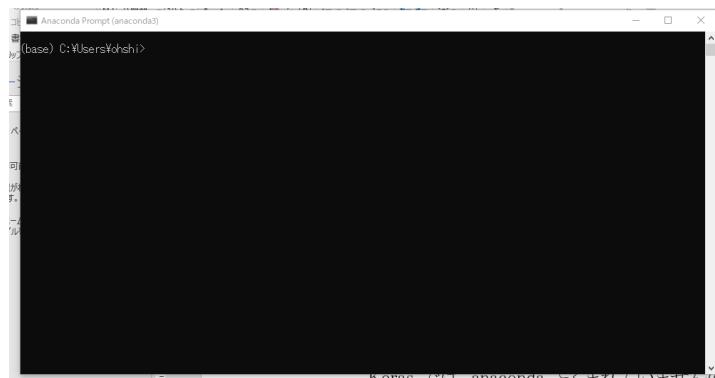
③作成コード「12桁の半角数字」を担当者に伝えてください。

(付記) Github の Omake コードに関する説明

演習でもちいる Github のコードには Keras をもちいた機械学習コードが含まれています。これは、ディープラーニングの全結合層をもちいることのできるモジュールとなります。ここでは、手書き文字(minist)や肺がん患者・健常者の識別などのコードが含まれています。

Keras では、anaconda ふくまれていませんので、別にモジュールをインストールする必要があります。手順は以下の通り。

①Anaconda Prompt を立ち上げます。



②次に、「pip install tensorflow」を打ち込んでリターンします。(Windows10 の場合)

インストールに成功すれば、Omake コードの「Omake_bank_datafolder (Keras).py」等を実行してください。

手書き文字では、Open-CV が必要です。こちらも anaconda ふくまれていませんので、別にモジュールをインストールする必要があります。手順は Keras 同様に、Anaconda Prompt を立ち上げた後に「python -m pip install -U opencv-python」をうちこみます。

```
(base) C:\Users\ohshi>python -m pip install -U opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.4.58-cp38-cp38-win_amd64.whl (35.1 MB)
    |████████| 35.1 MB 6.4 MB/s
Requirement already satisfied, skipping upgrade: numpy>=1.17.3 in c:\users\ohshi\python\lib\site-packages\numpy (1.19.2)
Installing collected packages: opencv-python
  Attempting uninstall: opencv-python
    Found existing installation: opencv-python 4.5.1.48
    Uninstalling opencv-python-4.5.1.48:
      Successfully uninstalled opencv-python-4.5.1.48
Successfully installed opencv-python-4.5.4.58
```

うまくいかないケースもありますが、担当者に相談いただければ対応いたします。