

Python プログラミング実践—機械学習—

今回はAI（人工知能）の一種である機械学習をプログラムします。機械学習とは、データから特徴を捉えて法則化（モデル化）し、その法則（モデル）を自動化することで、予測などをすることができます。

機械学習の手法は、分類をしていく決定木、脳の神経細胞に見立てたニューラルネットワーク（生物図表に掲載されています）など様々な方法です。今回はロジスティック回帰分析という手法を使用したとてもシンプルなプログラムで行います。

手順の概要は、

- 1 ライブラリをインポートする
- 2 学習させるデータを読み込む
- 3 基本統計量を算出する
- 4 データを学習データと検証データに分ける
(学習データは皆さんの勉強と同じ、検証用データをテストと同じです)
- 5 学習させてモデルを作成する
- 6 モデルの精度を確認する (精度が低い場合はモデルを改めて作成する)

課題

生徒100名の「英検の合否」を「勉強時間」「ゲーム時間」「通塾」から予測できるモデルを作成する

1. 必要なモジュールのインポート

複雑な計算処理をするためには、モジュールと呼ばれる予め作られたプログラムをインポートする。今回は、

- numpy・・・様々な計算に対応するためのもの
- pandas・・・データを計算しやすいようにデータを整理するためのモジュール
- sklearn.linear_model・・・ロジスティック回帰分析のためのモジュール
- sklearn.model_selection・・・データを学習用と検証用に分割するためのモジュール

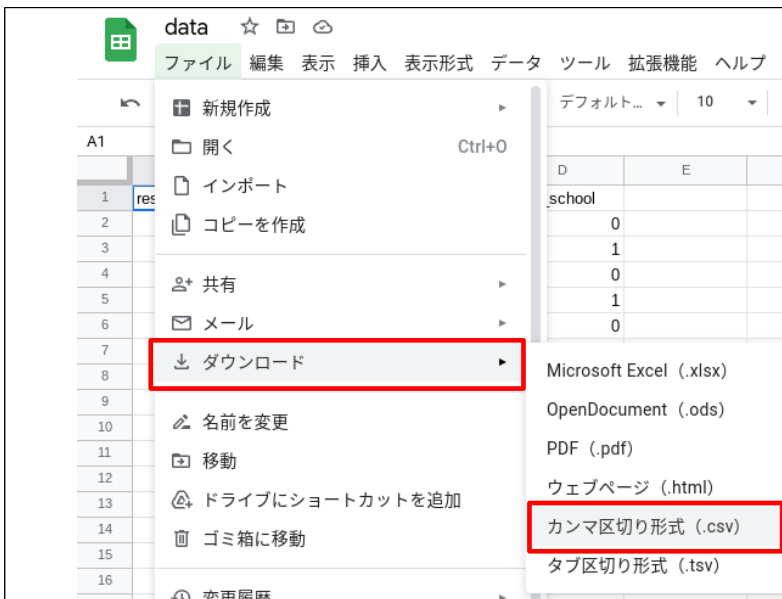
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

2. データの取得（学習させるデータを読み込む）

今回は配信されたデータ「data」を使用する。なお、プログラミングで取得する数値データのファイルはcsvファイル形式でなければならない。

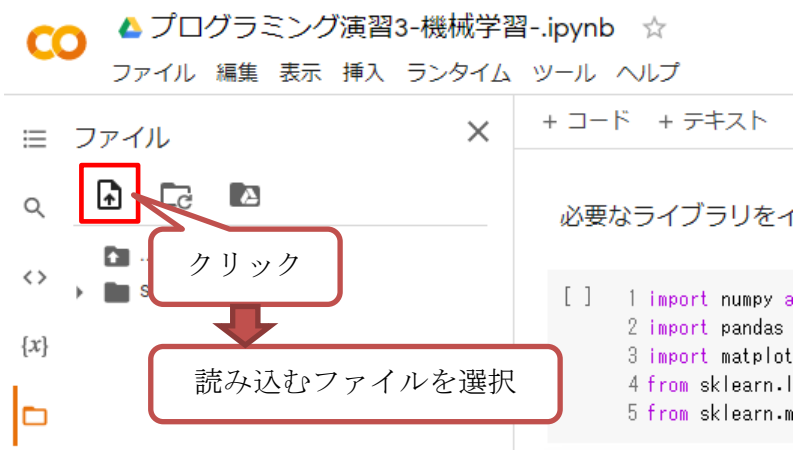
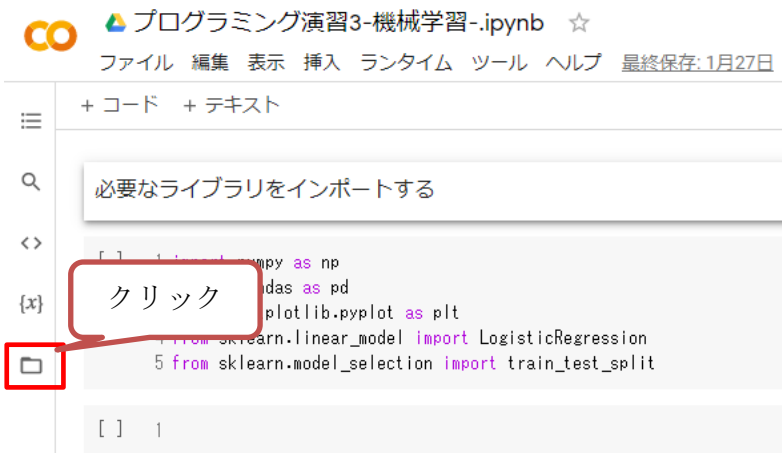
①データを開く

②「ファイル」→「ダウンロード」→「カンマ区切り形式(.csv)」でダウンロードする

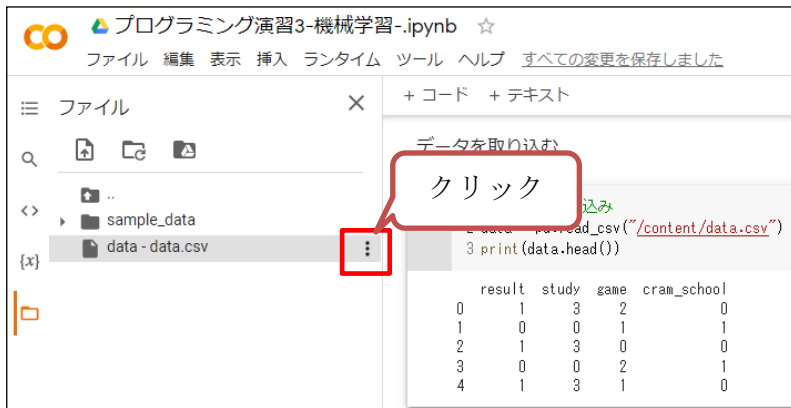


③データはchromebookの「ファイル」→「ダウンロード」にある

④データを jupyter notebook に読み込む



⑤読み込んだファイルのパスを取得する



⑤ファイルのパスを、pd.read_csv(“ ”)の” ” の間にペーストする

ペーストするときは「ctrl」キー＋「v」キーを使用する

*data = pd.read_csv(“ファイルのパスを「ctrl」キー＋「v」キーでペーストする”)

data.head() で最初の5行が出力される

データの説明

- resultは合格結果で、1は合格、0は不合格
- studyは勉強時間
- gameはゲーム時間
- cram_schoolは塾のことで、1は塾に通っている、0は塾に通っていない

```
#データの取り込み
```

```
data = pd.read_csv("/content/data - data.csv")
print(data.head())
```

```
result study game cram_school
0      1     3   2         0
1      0     0   1         1
2      1     3   0         0
3      0     0   2         1
4      1     3   1         0
```

3. 基本統計量を算出する

データの基本統計量を算出する

```
print(data.describe())
```

	result	study	game	cram_school
count	100.000000	100.000000	100.000000	100.000000
mean	0.620000	1.840000	1.040000	0.500000
std	0.487832	1.212061	0.803025	0.502519
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000
50%	1.000000	2.000000	1.000000	0.500000
75%	1.000000	3.000000	2.000000	1.000000
max	1.000000	3.000000	2.000000	1.000000

4. データを学習データと検証データに分割する

ロジスティック回帰分析は「 $Y = aX_1 + bX_2 + cX_3 + d$ 」に当てはめて分析する手法である。Yを目的変数、X1~X3を説明変数という。次の手順で、データを分析できる形式に加工する。

手順
①目的変数を合否結果（result）、説明変数をそれ以外の項目（勉強時間、ゲーム時間、通塾）とする。

- ・`X=data.drop(“ ”)`で、“ ”のデータを除外する。つまり、Xは勉強時間、ゲーム時間、通塾になる。なお、`axis=1`は「列方向」という意味である。

- ・`Y=data[“ ”]`で、Yは“ ”の項目になる。

②データを学習データと検証データに、50%ずつ分ける。

- ・`X_train`は学習データの説明変数、`X_test`は検証データの説明変数、`Y_train`は学習データの目的変数、`Y_test`は検証データの目的変数である。

- ・分割する割合は`test_size`で決まる。特に指定しなければ、乱数を発生させてデータをランダムで選択して学習データと検証データに分けることになるが、実行するたびに精度が変わることになる。それを避けるために、`random_state`で固定させることができる。今回は0としているが、数字は何でもよい。

```
#目的変数(Y)と説明変数(X)に分ける
```

```
X = data.drop("result", axis=1)
```

```
Y = data["result"]
```

```
#YとXを学習データと検証データに50%で分割する
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.5, random_state=0)
```

4. データを学習データと検証データに分割する

モデルを作成する

```
model = LogisticRegression() # ロジスティック回帰モデルを作成
model.fit(X_train, Y_train) # ロジスティック回帰モデルの重みを学習

LogisticRegression()
```

5. モデルの精度を確認する

精度を検証するためのモジュールをインポートして算出する。Train scoreは学習データで作成されたモデルのスコアで、92%の精度ということになる。Test scoreは、作成されたモデルを訓練データで検証したときの精度である。結果より、このモデルの精度は84%ということになる。

```
モデルの精度 (スコア)
from sklearn.metrics import confusion_matrix
print('Train score: {:.4f}'.format(model.score(X_train, Y_train)))
print('Test score: {:.4f}'.format(model.score(X_test, Y_test)))

Train score: 0.9200
Test score: 0.8400
```

6. サンプルデータを入力し予測 (判定) させる

せっかくなので、作成したモデルを使って予測を試してみる。sample = np.array([])にデータを入れ、サンプルを作成する。 []内の左から勉強時間 (0~3)、ゲーム時間 (0~3)、通塾 (0~1) を入力する。

このデータ形式では処理できないので、reshapedで変更し、model.predictで予測する。予測結果は、[1]で合格、[0]で不合格である。

```
sample = np.array([ 2, 1, 0 ])
reshaped_sample = sample.reshape([1, 3])
sample_predicted = model.predict(reshaped_sample)

print(' 合否予測:', sample_predicted)

合否予測: [1]
```

<注意点>

機械学習ではコンピュータが独自にモデルを作成するため、人間がそれを完全に理解することができないことが問題になっています。これをブラックボックス問題といいます。

また、モデルの精度が100%になることはほぼないです。つまり、完璧なAIを作成することはほぼ不可能です。AIの予測を鵜呑みしないことを意識する必要があります。

これらのことを踏まえて、AIと共生する社会を作る必要があります。