

# 機械学習 入門：予測（回帰）

広島大学 AI・データイノベーション教育研究センター  
滑川 裕介

# 目標

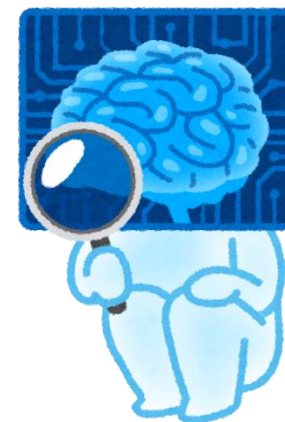
機械学習を用いた予測（回帰）が実行できるようになる。

今回、紹介すること

- 予測（回帰）に関わる機械学習の概観
- Python による機械学習を用いた予測

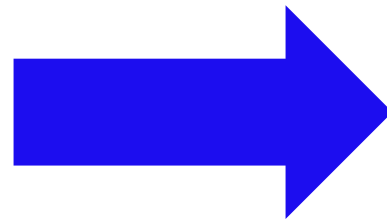
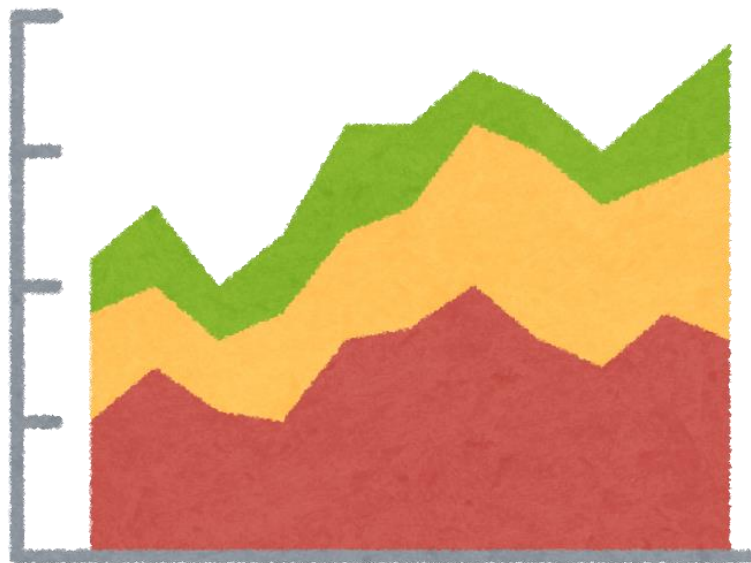
キーワード

機械学習、予測（回帰）、線形回帰

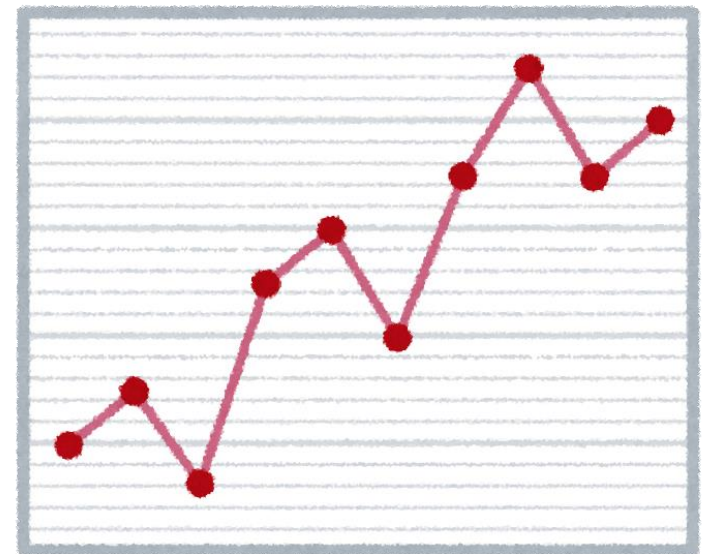


# こんなことは、ありませんか？

将来のことがある程度予想できればと考えたことは、ありませんか？  
完全な予知は難しいですが、需要予測・渋滞予測など機械学習により大幅に精度が向上した事例があります。



AI予測



# 機械学習による予測

機械学習とは人工知能の一種であり、データの学習により自動的に性能を改善する手法です。機械学習を用いることで、データの分類・予測（回帰）ができます。膨大な学習データが利用できるようになり機械学習の予測精度が高まりました。

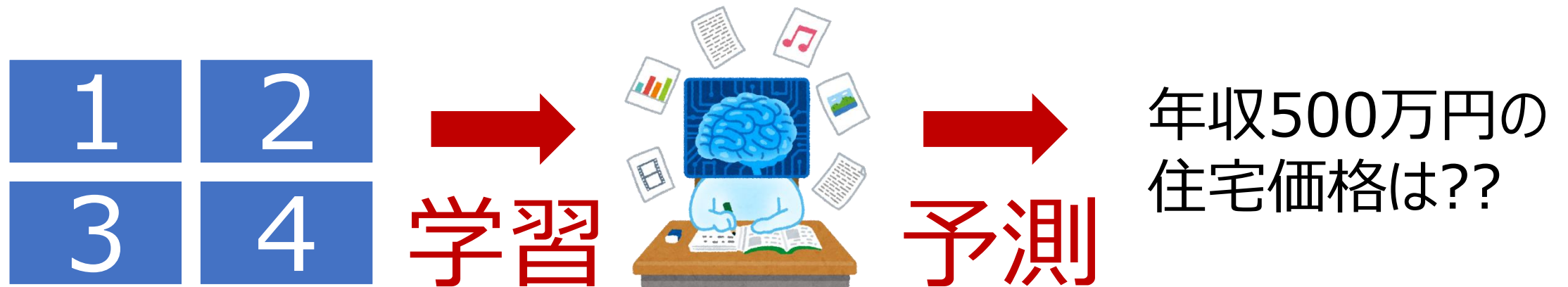
分類

予測



# 例題：年収と住宅価格の関係を予測する

実際にデータを学習させ、予測してみましよう。



今回は scikit-learn (サイキット ラーン) という Python の機械学習ライブラリを使用します。Google Colab などで次ページのプログラムを実行してみてください。

# 例題の Python コード

予測結果  
おおむね正しく  
予測できている

```
# ライブラリの読み込み
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pandas as pd

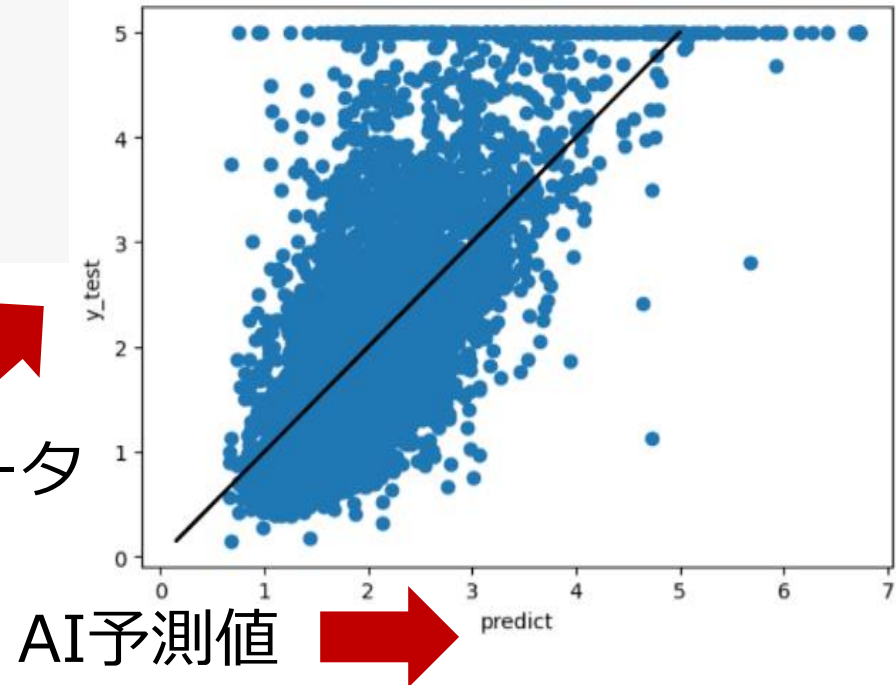
# 収入(dataの0列目)と住宅価格(target)データの読み込みと前準備
data = fetch_california_housing()
x_train, x_test, y_train, y_test = train_test_split(pd.DataFrame(data.data[:,0]), data.target)

# 訓練データの学習
model = LinearRegression()
model.fit(x_train, y_train)

# テストデータを用いた判定
predicted = model.predict(x_test)

# 判定結果の図示
import matplotlib.pyplot as plt
plt.xlabel("predict")
plt.ylabel("y_test")
plt.plot(y_test, y_test, color='black')
plt.scatter(predicted, y_test)
plt.show()
```

実際の住宅価格データ



# 例題の解説 1 : scikit-learn ライブラリ

# ライブラリの読み込み

```
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pandas as pd
```

scikit-learn (サイキット ラーン) は Python の機械学習ライブラリです。

練習用データセット、データ整形ツール、機械学習部分を読み込んでいます。



## scikit-learn

*Machine Learning in Python*

Getting Started

Release Highlights for 1.4

GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license



# 例題の解説 2 : 年収、住宅価格データ

```
# 収入(dataの0列目)と住宅価格(target)データの読み込みと前準備
data = fetch_california_housing()
x_train, x_test, y_train, y_test = train_test_split(pd.DataFrame(data.data[:,0]), data.target)
```

練習用データセットライブラリから、1990年カリフォルニア州の年収・住宅価格データを読み込んでいます。

ただし、そのままでは使いづらいため、データの変形および分割をしています。

## データ例

年収 [1万ドル]	住宅価格 [10万ドル]
8.3252	4.526
8.3014	3.585



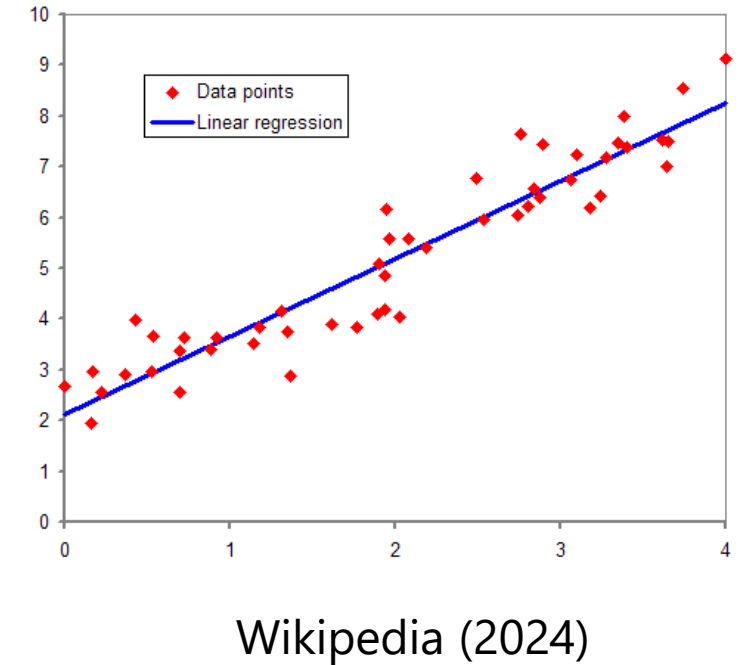
# 例題の解説 3 : 線形回帰

```
# 訓練データの学習
```

```
model = LinearRegression()  
model.fit(x_train, y_train)
```

例題では線形回帰（Linear Regression）を学習・予測に用いました。

他にも、多重回帰など、さまざまな手法があります。



線形回帰：

学習データ間に線形の関係を仮定し、出力を与える方法。

# 例題の解説 4 : matplotlib による図示

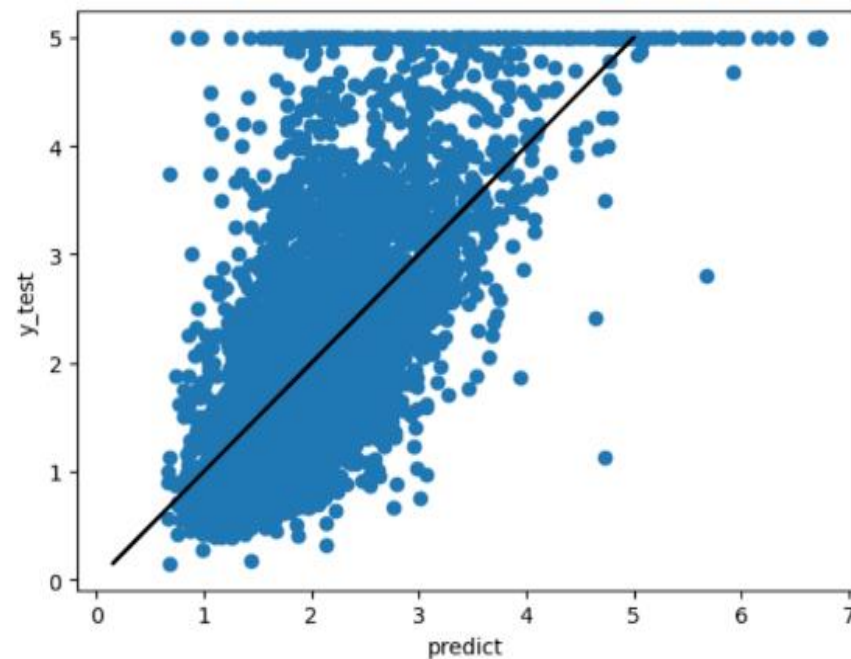
```
# 判定結果の図示
```

```
import matplotlib.pyplot as plt
plt.xlabel("predict")
plt.ylabel("y_test")
plt.plot(y_test, y_test, color='black')
plt.scatter(predicted, y_test)
plt.show()
```

機械学習で判定した結果を matplotlib ライブラリを用いて図示しています。

**matplotlib**

予測結果



# 例題の解説 5 : AI予測結果の解釈

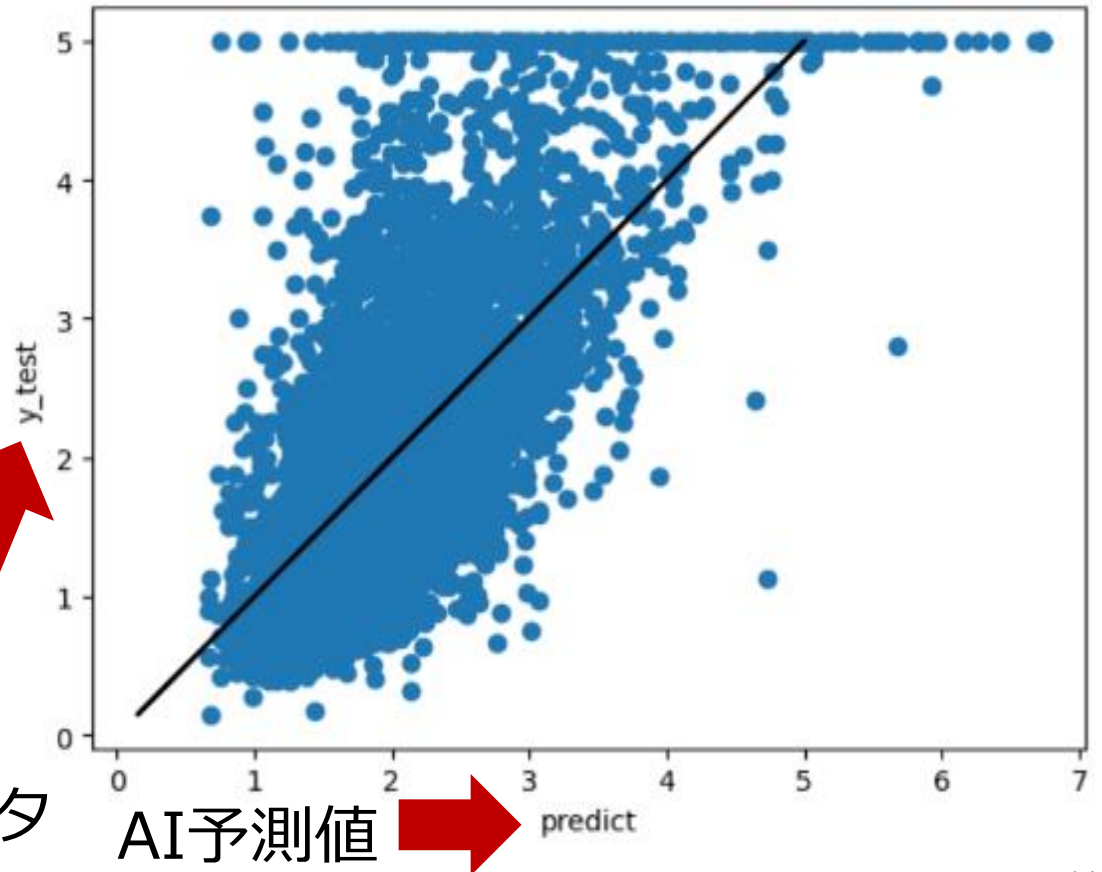
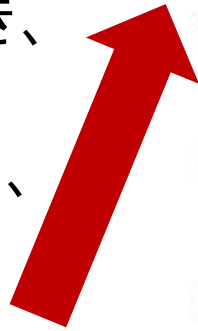
## 予測結果

機械学習で予測した結果は  
実際のデータ ( $y_{\text{test}}$ ) と  
おおむね一致しています。

ただし、完璧に一致している  
わけではありません。

- 原因 1 : 線形回帰は単純すぎ、  
多重回帰が必要
- 原因 2 : データに外れ値があり、  
処理が必要

実際の住宅価格データ



# 機械学習による予測の応用例

ここまで説明してきた機械学習による予測（回帰）はすでにさまざまな現場で利用されています。

- 需要予測：在庫管理、販売計画立案に利用
- 渋滞予測：物流の最適化に利用
- 場面予測：囲碁や将棋の盤面予測に利用
- 行動予測：犯罪者の行動予測に利用
- 感染予測：インフルエンザなどの流行予測に利用

注) 学習に基づくため、前例の無い事の予測は苦手

# まとめ

機械学習を用いた予測（回帰）  
について解説しました。



Python 機械学習ライブラリ scikit-learn を使用することで、  
比較的簡単に機械学習を用いた予測（回帰）ができます。

今回説明した機械学習による予測は、さまざまな現場で使用  
されており、今後も幅広い分野で応用されると考えられます。