

3章-1 線形回帰

回帰とは

ある値（目的変数）を別の値（説明変数・特徴量）で説明するタスクのことを 回帰 と言います。

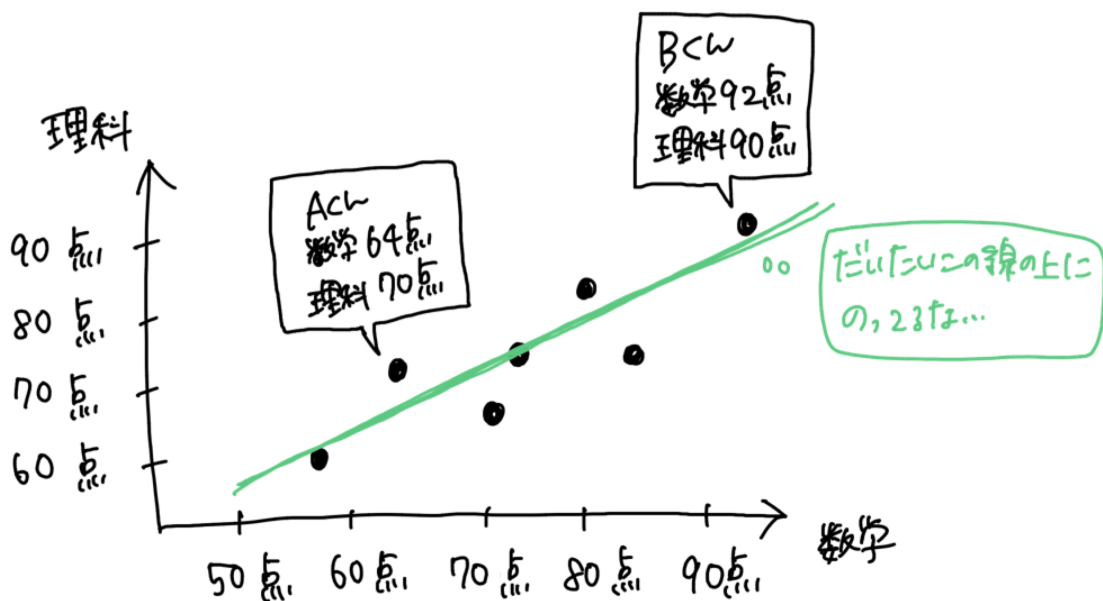
（回帰具体例）

「身長」を「足のサイズ」から推論します。

「理科の点数」を「数学の点数」から推論します。

線形回帰とは

線形回帰はデータの傾向を要約し、直線を当てはめて分析する手法です。



- 当てはめる直線は以下の式のように表されます。

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_px_p$$

y：目的変数

x1～xp：説明変数（=特徴量）

データを最もよく説明するa0,a1～apを求め、その式を使って推論を行います。

- 説明変数が1変数のときを 単回帰分析、2変数以上のときを 重回帰分析 といいます。

単回帰分析：「家賃」を、「部屋の広さ」だけで推論します。

重回帰分析：「家賃」を、「部屋の広さ」「築年数」「駅からの近さ」など複数の要素から推論します。

実践では複数の特徴量から回帰分析を行うことが多いです。

重回帰分析 のコードを実際に動かしてみましょう。

```
from sklearn.linear_model import LinearRegression
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split

# Bostonデータセットを読み込む
boston = load_boston()

# Bostonデータセットを見ていく
print(type(boston))
print(boston.keys())
print(boston['data'])
print(boston['target'])
print(boston['feature_names'])
print(boston['DESCR'])

X = boston.data
y = boston.target

print(type(X))
print(type(y))
print(X.shape)
print(y.shape)

# Bostonデータセットを学習データセットとテストデータセットに分割
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
```

```
random_state=123)

# X_trainデータをデータフレーム素材で見よう
import pandas as pd
df_Xtrain = pd.DataFrame(X_train, columns = boston.feature_names)
df_Xtrain

# 線形回帰をインスタンス化
bos = LinearRegression()

# 学習
bos.fit(X_train, y_train)
y_pred = bos.predict(X_test)

# 予測結果を見る
print(y_pred)

# 予測結果がどのくらい当たっているか見てみよう
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.scatter(y_pred, y_test)
ax.plot((0, 50), (0, 50), linestyle='dashed', color='limegreen')
ax.set_xlabel('predicted value')
ax.set_ylabel('actual value')
plt.title('boston house value')
plt.show()
```