

しばらくおまちください

<https://wuff-ccrknyhaca-de.a.run.app/>

にテキストを書くとこの画面に流れます

(#event/workshop/machine-learning/links にリンクが貼ってあります)

機械学習講習会 第一回

- 「学習」

traP アルゴリズム班 Kaggle部
2023/06/20

はじめに

- ✓ 自分で設計したニューラルネットワークを学習させて、問題を解く。

全7回 + 演習

毎日 20:00~ (初回のみ21:00~)

第1回 6/20: 学習

第2回 6/21: 勾配降下法

第3回 6/23: 自動微分とPyTorch

第4回 6/27: ニューラルネットワークの構造

第5回 6/28: ニューラルネットワークの学習と評価

第6回 6/30: ニューラルネットワークの実装

第7回 7/04: ニューラルネットワーク発展

この講習会で扱うこと、扱わないこと

機械学習は非常に広大な分野 ⇨ 全7回ではちょっと限界がある

今回の講習会ではディープラーニングについて扱います

- ツールを触るだけで原理は全然やらない
- 原理をやるだけで全然使えない

にならぬようにどちらもバランス良くやります

全体の流れ

第1回: 学習

第2回: 勾配降下法

第3回: 自動微分とPyTorch

第4回: ニューラルネットワークの構造

第5回: ニューラルネットワークの学習と評価

第6回: ニューラルネットワークの実装

第7回: ニューラルネットワーク発展

Pythonを使います

自分のPCにインストール or
Google Colaboratoryなどを利用

- 第二回までに実行できるようにしておいてください
- <https://md.trap.jp/eRvsfsf3RhmoNNwvWlku7A>
(linksチャンネルを見てください！)



Pythonを使った初步的なプログラミング

- if文, for文, 関数 など
- 外部パッケージの利用

(そこまで高度なことはやりません)

数学の初步的な知識

- 基本的な行列の演算や操作
- 基本的な微分積分の知識(偏微分など)



abap34

- 情報工学系B2
- Kaggle部部長

運営を手伝ってくれた人

内容の議論・チェックなど

- 23B kobakosくん
- 22M YumizSuiさん
- 22M idatenさん
- 22M Silviaseさん

ありがとうございます🙏



Wuff

- テキストを書くとこの画面に流れます
- 匿名なのでお気軽にどうぞ

#workshop/machine-learning/sodan

- [traQの相談チャンネル #event/workshop/sodan](#)

ありとあらゆるところにお気軽になんでも書いてください！

timesに書くときはしれっと「機械学習講習会の...が...」みたいにワードを入れてもらえれば拾います！

前提として、大変

1. 講義パートで引っかかったら秒で書き込みましょう！(traQでもwuffでも **OK**)
2. 演習パートでは、5分わからなかつたら質問してもらって大丈夫です！

もっと大前提として、質問は迷惑ではないです！
(むしろ聞いてもらえているということなので、うれしい)

がんばりましょう

(ここだけの話、機械学習はめちゃくちゃおもしろい)

全7回、がんばりましょう！！

第一回：學習

今日の目次

1. 機械学習 or AI?
2. 二つの変数の関係を予想しよう ~アイスの売り上げ予測~
3. モデルとパラメータ
4. 「いいモデル」とは？ 損失関数の登場

今日の目標

機械学習モデルを構築する上での基本的な用語を整理して、「学習」ということばをきちんと説明できるようになる。

大流行、生成系AI

- ChatGPT  自然な対話応答
- Stable Diffusion  お絵描き
- GitHub Copilot  プログラミング

etc...

機械学習 or AI ?

- AI(人工知能)
「人間っぽい知能」を実現しようとする分野・あるいは知能そのもの
- 機械学習(Machine Learning, ML)
様々な情報から「学習」をして動作するアルゴリズム
人工知能の一つのかたちと見られることが多い



機械学習 ⊂ 人工知能
(\leftrightarrow スーパーカー ⊂ 地上をめっちゃ速く走る)

ここでは一つの定義を紹介しましたが、実際この二つの言葉に明確に定義や合意があるわけではないです。
なので人工知能か人工知能でないか、機械学習かそうではないか、みたいな議論はやや不毛そうです。

学習ってなに？

- 機械学習(Machine Learning, ML)
様々な情報から「学習」をして動作するアルゴリズム

↑ 学習って何？

今日のテーマ:

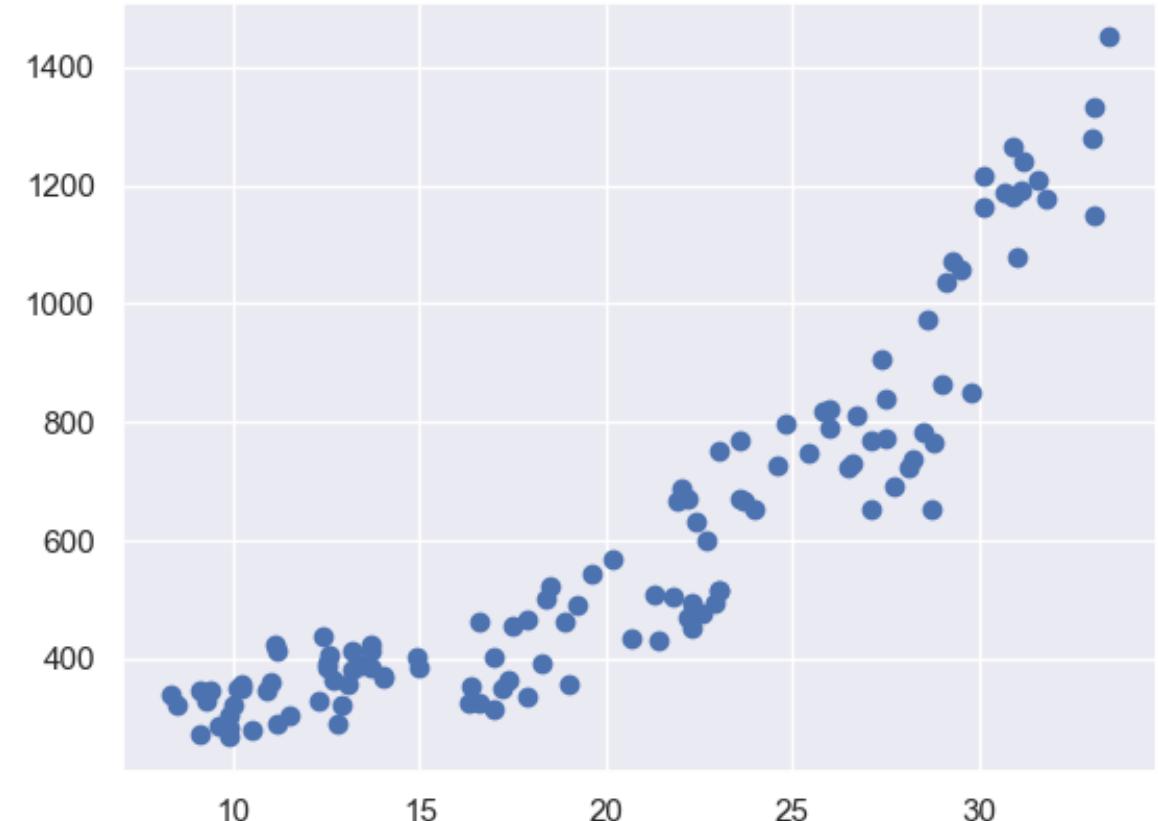
-  「学習」を説明できるようになる

「気温」と「アイスの売り上げ」

- 気温↑ → 売れそう
- 気温↓ → 売れなさそう

「アイスの売り上げ」は「気温」からある程度わかりそう？

⌚ < 来月の売り上げが予想できたらどのくらい牛乳仕入れたらいいかわかつて嬉しいな。



データは

<https://okumuralab.org/~okumura/stat/160118.html>
から引用

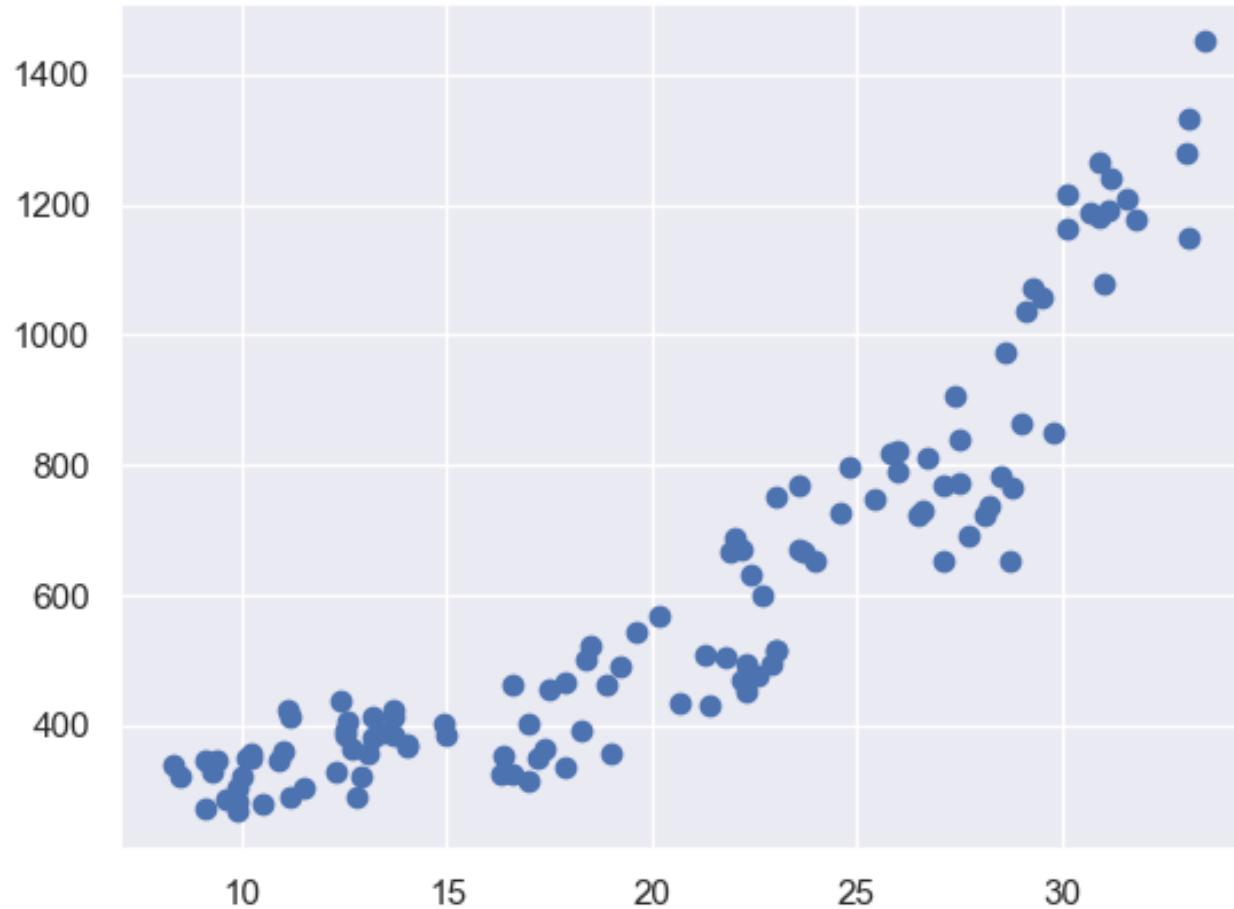
アリスの売り上げを予測する

⌚ <なんか来月の予想平均気温30度って気象庁が言ってたな。



⌚ < !!!!!

アリスの売り上げを予測する



⌚ < 過去に30°Cのときは...

アリスの売り上げを予測する

一番簡単な方法: 過去の全く同じ状況を参照する

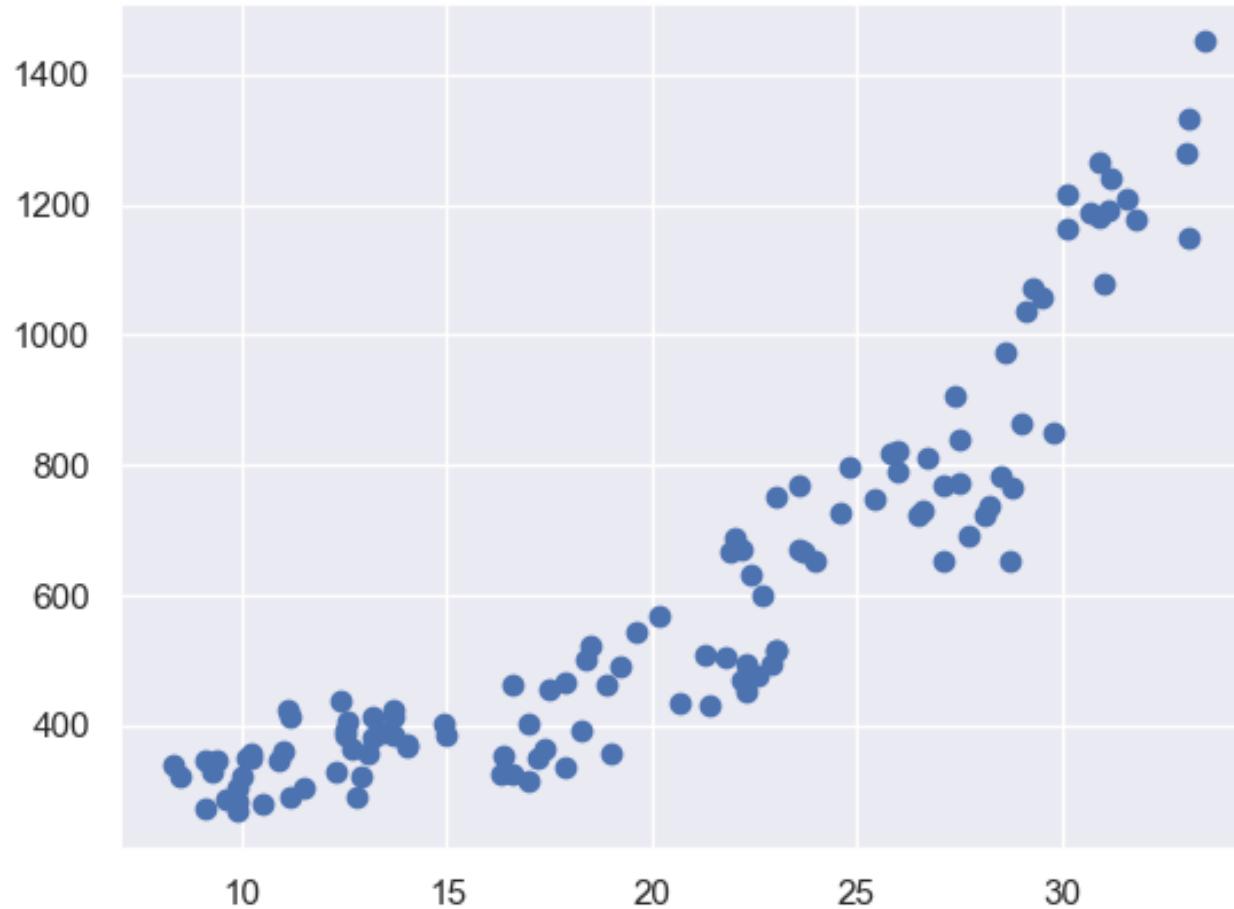
🐻 < ガハハ ! これでアイスの売り上げを予測するAIの完成や !



< そのまた来月の予想平均気温は40°Cです。

🐻 < !?

アリスの売り上げを予測する



🐶 < 40°Cないやんけ

アリスの売り上げを予測する

「予測」 ... 入力から出力を求める

今回は、「入力: 気温」 → 「出力: アイスの売り上げ」

- ✓ 入力は知ってるものだけとは限らない

 ← こいつが本当にやらなくてはいけなかつたことは...

売り上げ = $f(\text{気温})$ となる関数 f の推定

このような、入力データを受け取り結果を返す f をモデルと呼びます。
(厳密に数学的な意味の関数である必要はない)

線形回帰

売り上げ = $f(\text{気温})$ となる関数 f が、

$$f(\text{気温}) = a \times \text{気温} + b$$

のかたちで表させると仮定して、 a と b を求める。

例) $a = 20, b = 100$ なら

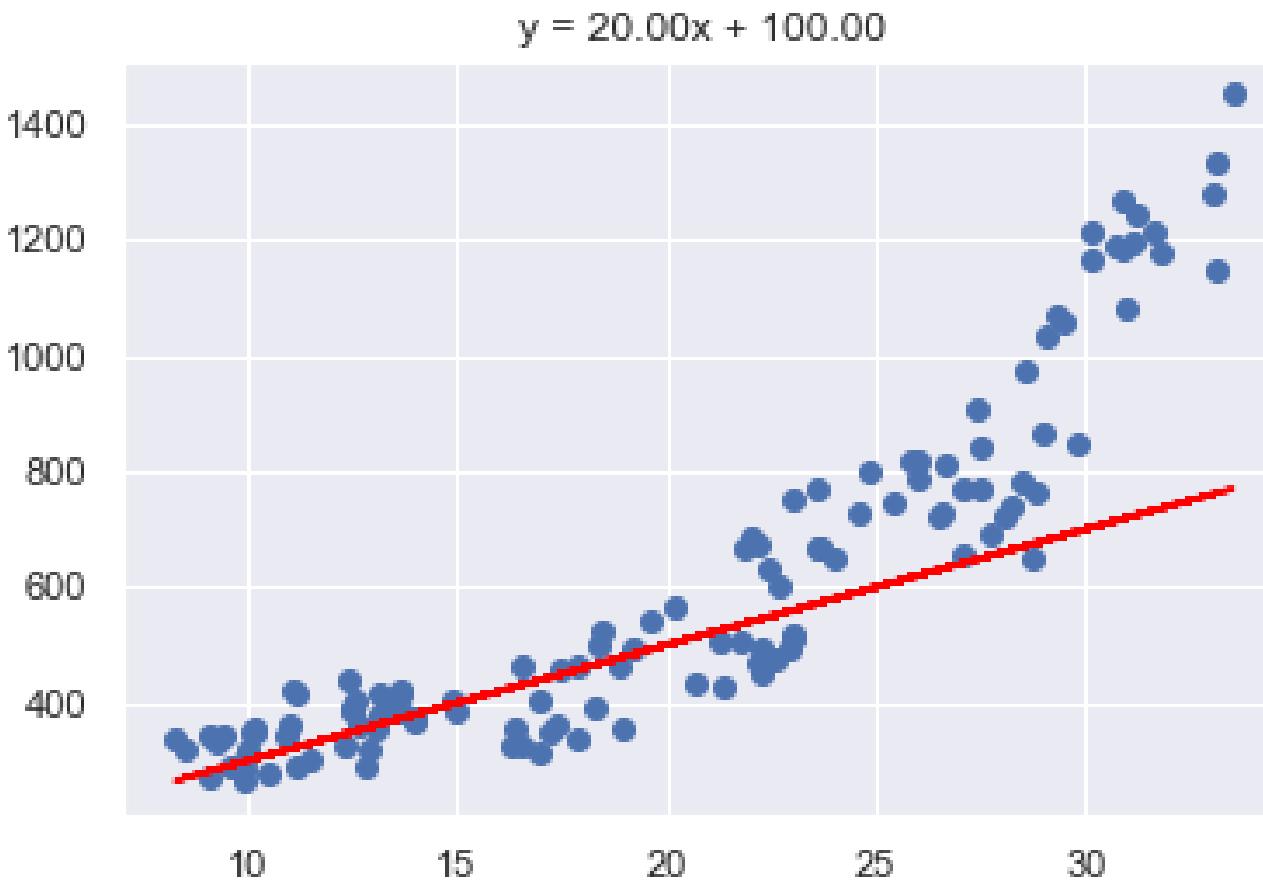
$$\text{アイスの売り上げ} = f(\text{気温}) = 20 \times \text{気温} + 100$$

f が入力変数の線形結合で表せると仮定して、

$y = f(x)$ の関係を求める作業を「線形回帰」といいます。

線形回帰

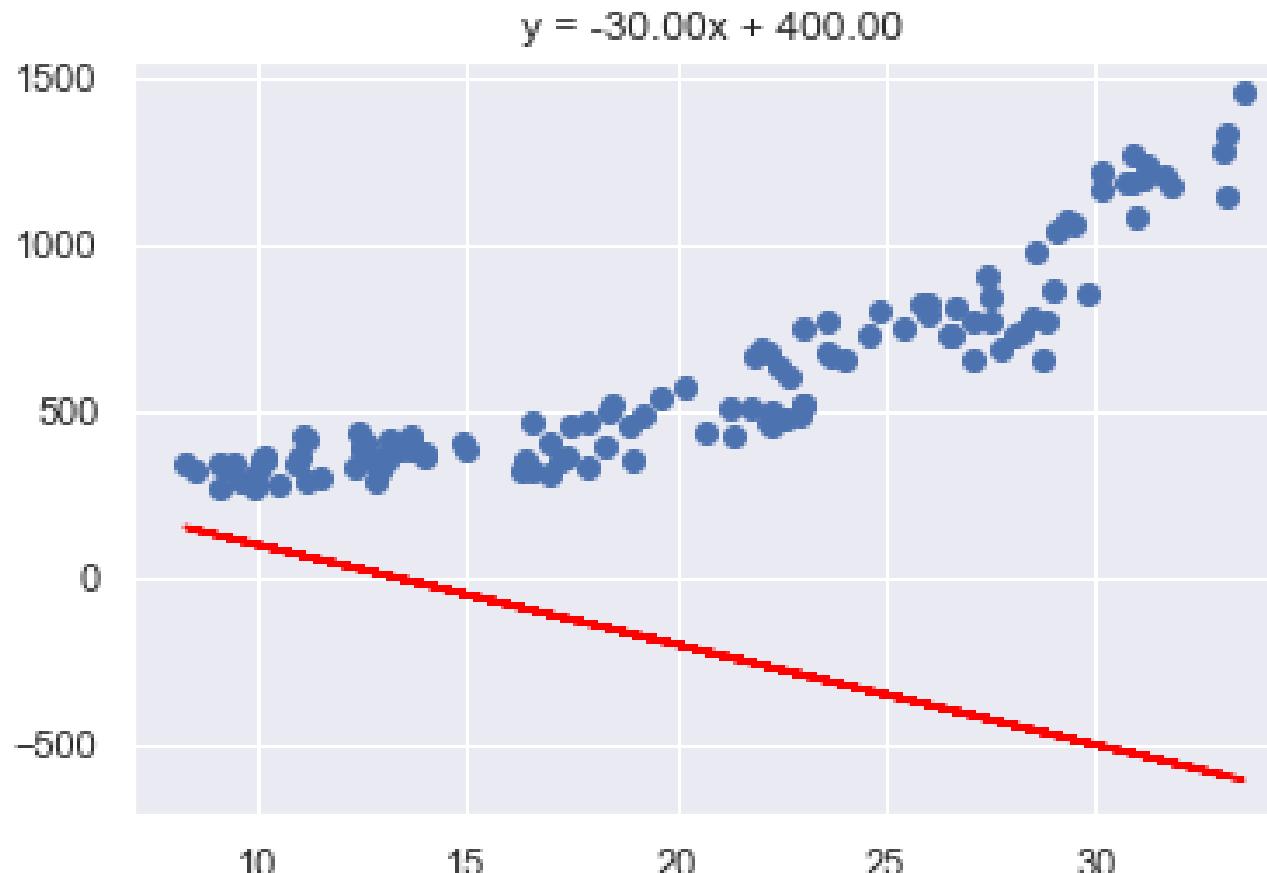
ためしてみる



< わるくないね

線形回帰

ためしてみる2 ... $a = -30, b = 400$



< 帰れ

パラメータ

a, b を変えることでモデルの具体的な形が変わった！

このように各モデルが固有に持つてモデル自身の性質を定める数を、「パラメータ」という。



✓ 入力は知ってるものだけとは限らない

← こいつが本当にやらなくてはいけなかったことは...

売り上げ = $f(\text{気温})$ となる関数 f の推定

は、 f の構造を決めておけば...

「 f の推定 \leftrightarrow f のパラメータの推定」

ちょっとまとめ

- アイスの売り上げを予測するには、気温から売り上げを予測する「関数」を構築するのが必要であった。
- 今回は関数の形として $f(x) = ax + b$ (一次関数)を仮定して、「関数」を求めることにした。
- この関数は、パラメータとして a, b をもち、 a, b を変えることで性質が変わるのがわかった。
- a, b を定めることで具体的に関数が定まる。

休憩します！！

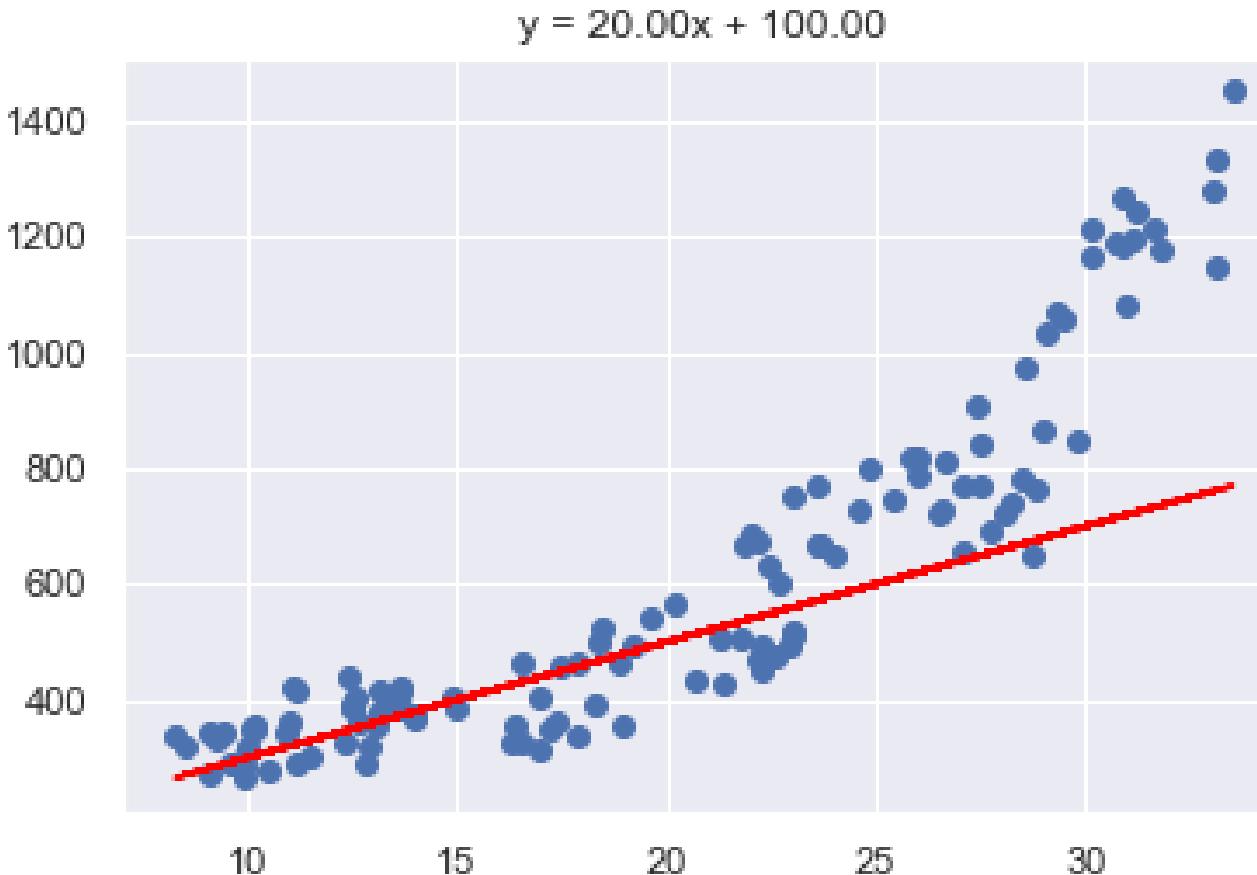
ちょっとまとめ

- アイスの売り上げを予測するには、気温から売り上げを予測する「関数」を構築するのが必要であった。
- 今回は関数の形として $f(x) = ax + b$ (一次関数)を仮定して、「関数」を求めることにした。
- この関数は、パラメータとして a, b をもち、 a, b を変えることで性質が変わるのがわかった。
- a, b を定めることで具体的に関数が定まる。



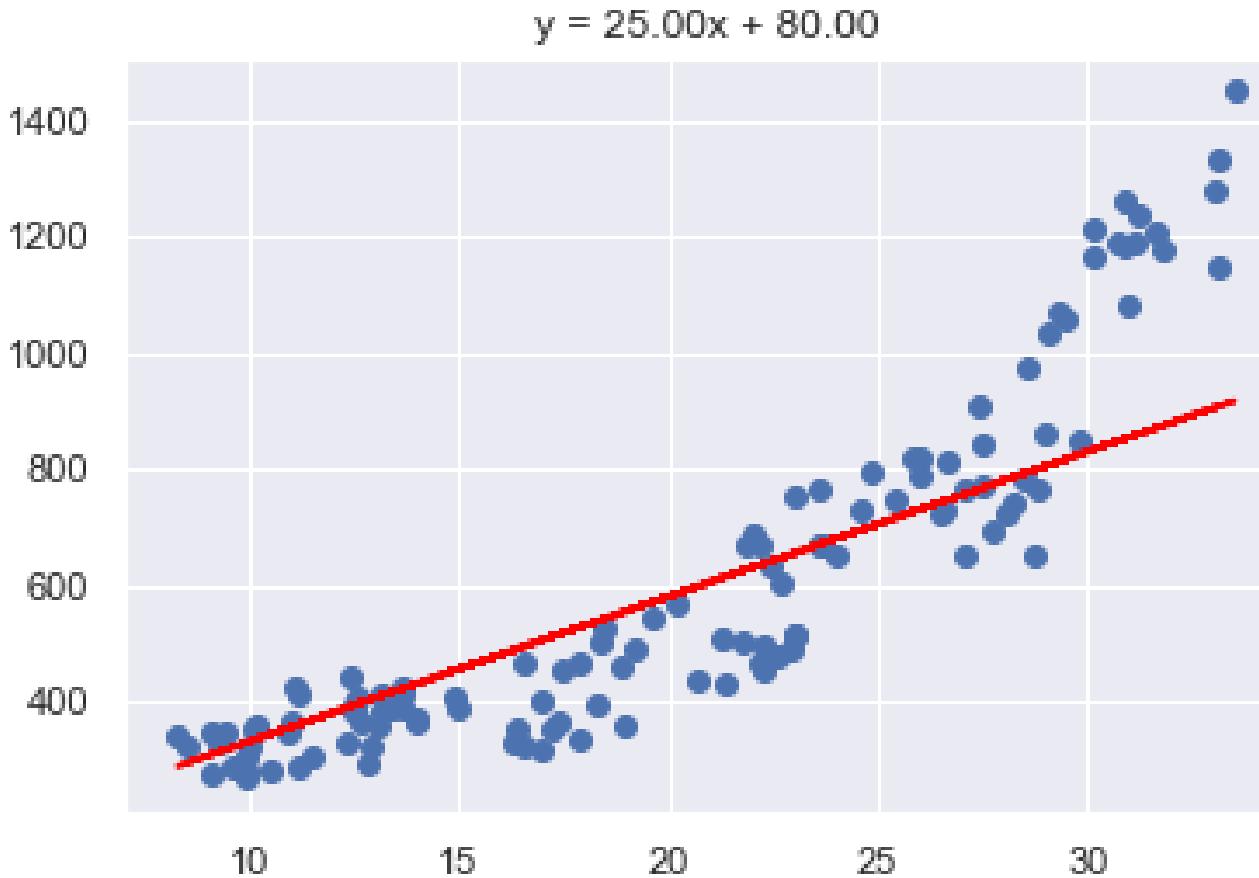
a, b を決めよう！

a, b の決め方



🐶 < わるくないね

a, b の決め方



🐶 <ちょっといい？

a, b の決め方: 損失の導入

「良さ」とはなにか？



「悪くなさ」



「悪さ」とはなにか？



「データと予測の遠さ」！

平均二乗誤差(Mean Squared Error)

$$\frac{1}{n} \sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

損失関数

「悪さ」の指標を「損失関数」という

学習 ...

「損失関数」を最小にする f のパラメータを見つける

(どんな複雑な f でも共通！！！！)

「悪さ」を最小化するのではなく「良さ」を最大化すれば良くない?と思った人もいるかもしれないですね。
実はそれはものすごくいい疑問で、次回以降で明らかになっていきます。
ひとまずは一旦疑問として抱えておいてください。

注意

損失は何の関数？

- ✓ 各 x_i, y_i は変数みたいな見た目だが、損失関数を考えるときは定数

$$L(\text{??}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

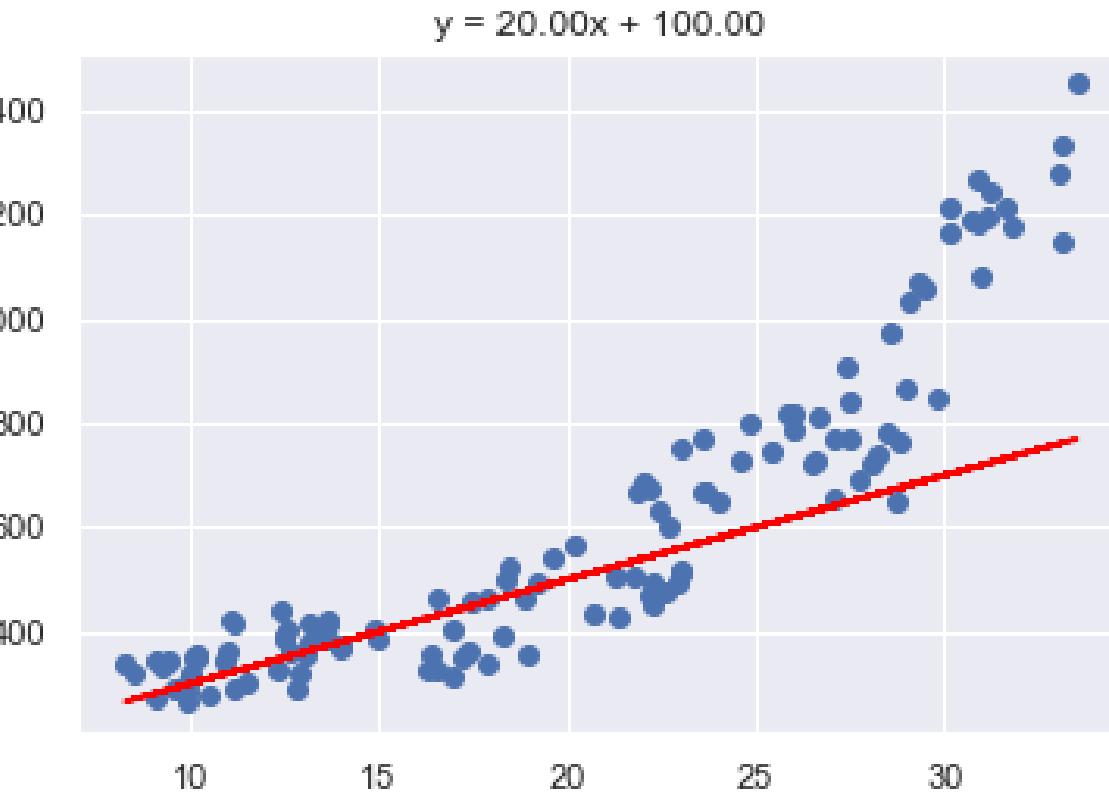


$$L(a, b) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - f(x_i; a, b))^2$$

(※当然例外もあります)

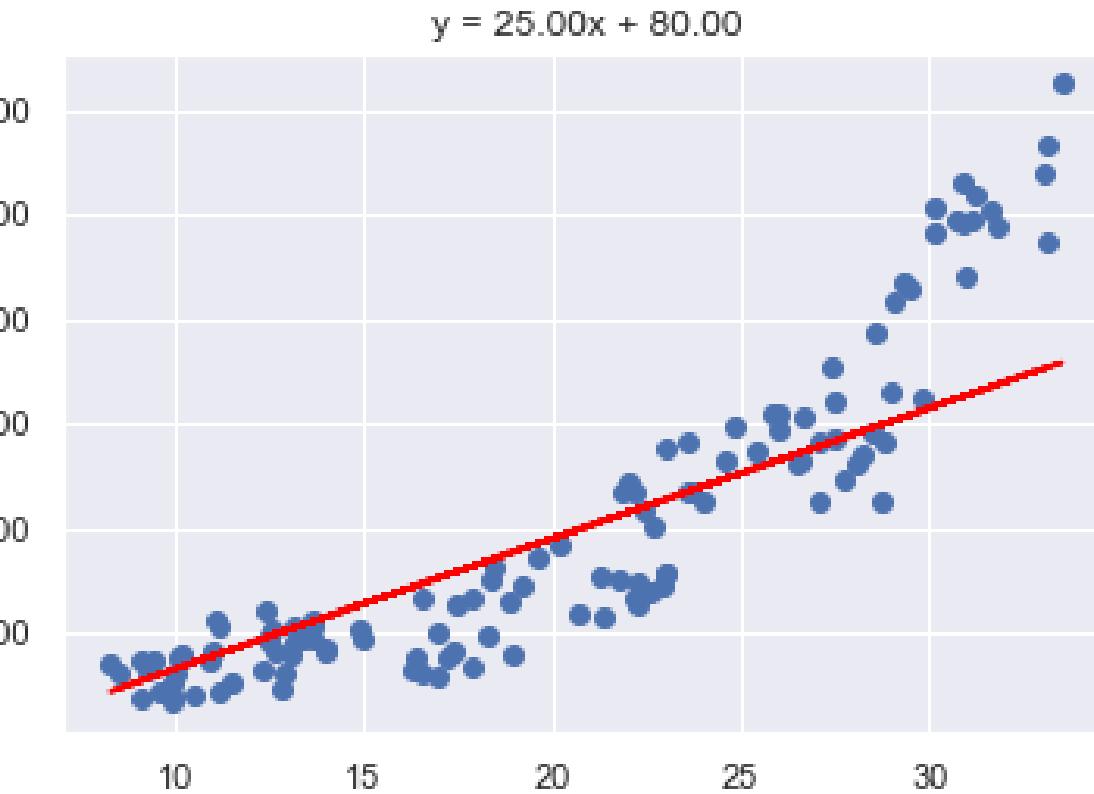
今回は...

$$L(20, 100) = 40268.55$$



今回は...

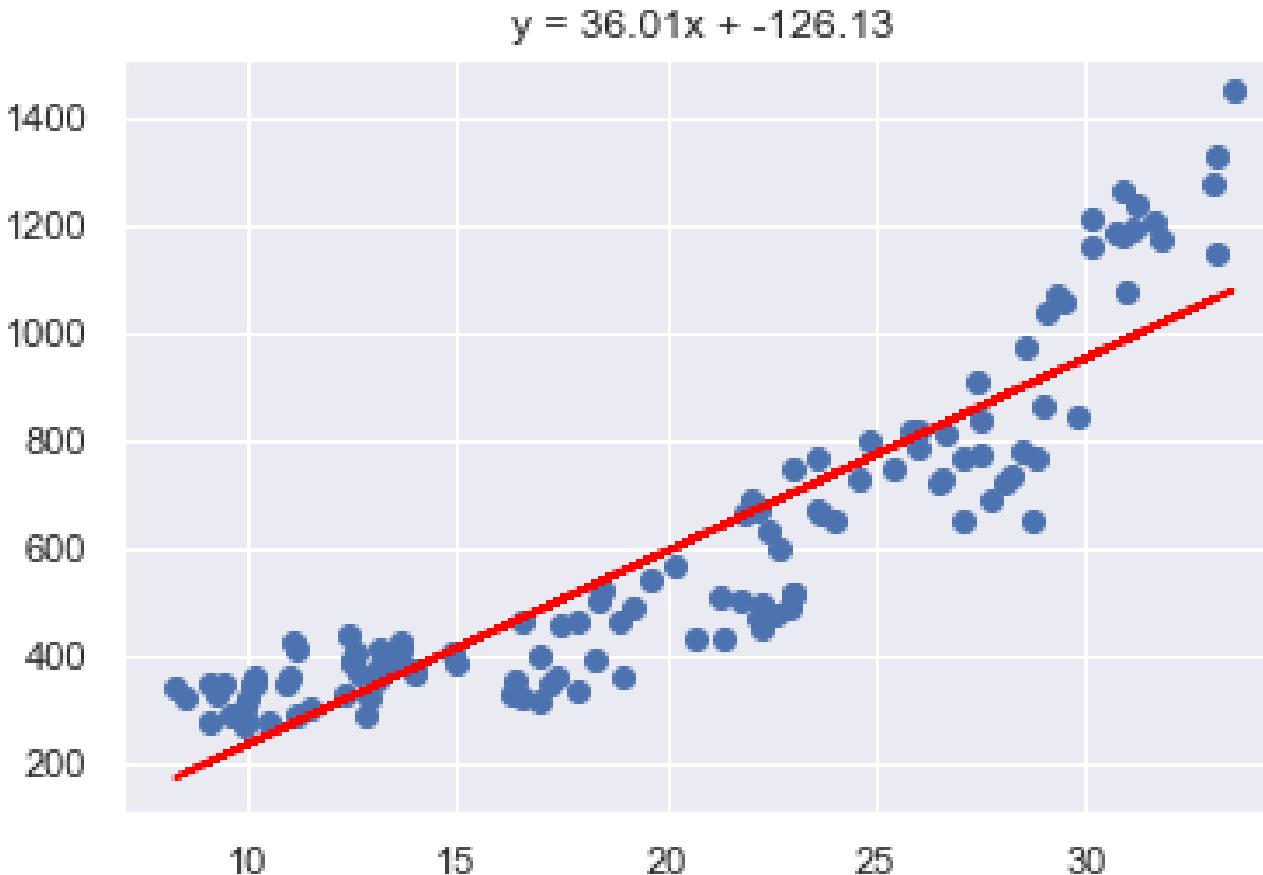
$$L(25, 80) = 23445.075$$



こっちの方がよかったです！

今回は...

$$L(a, b) = 16482.246499700002$$



いや

それ

どう

やったの

第二回：勾配降下法

まとめ

- アイスの売り上げを予測するには、気温から売り上げを予測する「関数」を構築するのが必要であった。
- 今回は関数の形として $f(x) = ax + b$ (一次関数)を仮定して、「関数」を求めることにした。
- この関数は、パラメータとして a, b をもち、 a, b を変えることで性質が変わるのがわかった。
- a, b を定めることで具体的に関数が定まる。
- このパラメータを決める基準として、「悪さ」の基準である損失関数を定めた。
- 損失関数の値を最小化する a, b を決めることを「学習」と呼ぶ。

演習1

1. 平均二乗誤差を計算するpythonのプログラムを実装してください。(不正な入力は来ないという想定でokです)

```
def mean_squared_error(y, pred):  
    # ...  
    # ここに書く  
    # ...  
  
    # アイスの売り上げは実際100個, 200個, 300個だった  
    y = [100, 200, 300]  
  
    # 150個, 220個, 300個と予測  
    pred = [150, 220, 300]  
  
    print('mse:', mean_squared_error(y, pred))
```

演習2

2. (1.)で実装した平均二乗誤差を使って、損失を a, b の関数として書いてください。

```
def loss(a, b):
    x = [25, 30, 35]
    y = [80, 90, 100]
    # ...
    # ここに書く
    # ...

print('loss:', loss(3, 1))
```

答え1

```
def mean_squared_error(y, pred):
    n = len(y)
    s = 0
    for i in range(n):
        s += (y[i] - pred[i])**2
    return s / n

y = [100, 200, 300]
pred = [150, 220, 300]

print('mse:', mean_squared_error(y, pred))
# mse: 966.6666666666666
```

答え2

```
def loss(a, b):
    x = [25, 30, 35]
    y = [80, 90, 100]
    pred = []
    for x_i in x:
        pred.append(a * x_i + b)
    return mean_squared_error(y, pred)

print('loss:', loss(3, 1))
# loss: 17.666666666666668
```

リスト内包表記を使ったバージョン

```
def loss(a, b):
    x = [25, 30, 35]
    y = [80, 90, 100]
    pred = [a * x_i + b for x_i in x]
    return mean_squared_error(y, pred)
```