

# 機械学習講習会

[1] 「学習」

2024/06/24  
traP Kaggle班

はじめに

# この講習会のゴール

---

- ✓ 機械学習の基本的なアイデアを理解して、  
問題解決の手段として使えるようになる。

# おしながき

- 第1回 | 学習
- 第2回 | 勾配降下法
- 第3回 | 自動微分
- 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 | PyTorch による実装
- 第7回 | 機械学習の応用、データ分析コンペ

# 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

今回の講習会ではとくに**ディープラーニング**についてメインに扱います

- 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

にならない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

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 PyTorch による実装

第7回 機械学習の応用、データ分析コンペ

## 最終的には...

- ✓ 機械学習の基本的なアイデアを説明できるようになる
- ✓ ライブラリに頼らず基本的なアルゴリズム, モデルを実装できるようになる
- ✓ PyTorch を使った基本的なニューラルネットの実装ができるようになる

# 使うプログラミング言語

---



## Python を使います

### 慣れている人へ

→ Jupyter Notebook と numpy, matplotlib, scipy, PyTorch あたりのライブラリを使えるようにしておいてください

### 慣れていない人へ

→ <https://abap34.github.io/ml-lecture/supplement/colab.html> をみて Google Colaboratory の使い方を覚えておいてください



# 使うプログラミング言語や前提知識など

## 1. Pythonを使った初歩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求めません **ググり力とかの方が大事**)

## 2. 数学の初歩的な知識

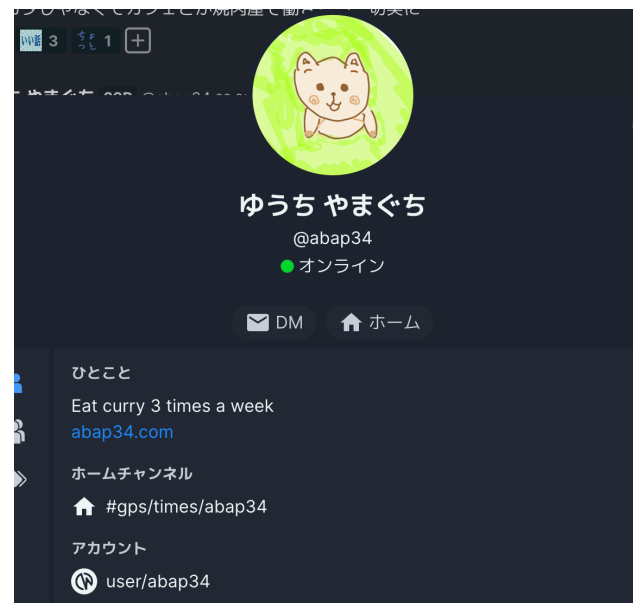
- 基本的な行列の演算や操作 (積、転置など)
- 基本的な微分積分の知識 (偏微分など)

(1年前期の(線形代数) + (微分積分のさわり) くらい)

# 著者のページ

## @abap34

- 情報理工学院情報工学系 B3
- Kaggle班班長
  - アルゴ班にも入ってます
- 趣味
  - 機械学習 🤖
  - 個人開発 ⚙️
  - 野球 ⚾️
  - 音楽 🎵



# 謝辞 .....

内容の議論・チェックなど

- 24D @YumizSui さん (ex-traP)
- 22M @idaten さん (ex-traP)
- 23B @Kobakos くん

ありがとうございます🙏

# 質問・相談など

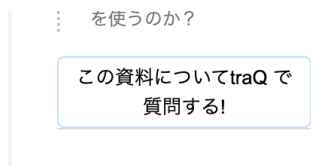
## 1. Wuff

- テキストを書くところの画面に流れます
- **匿名** なので気軽にどうぞ

## 2. #workshop/machine-learning/sodan

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

## 3. 各講習会資料に載ってる質問ボタン



← これを押すと投稿できます！

# 質問・相談など

---

前提として、 **大変**

1. 講義パートでは、5秒わからなかったら質問しよう  
(traQでもwuffでも **OK**)
2. 演習パートでは、5分わからなかったら質問しよう

# がんばりましょう

.....

(ここだけの話、機械学習はめっ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

# おしながき

.....

## 今日の目標

機械学習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。



# 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 or AI ?

.....

機械学習 で 人工知能 を実現

( $\leftrightarrow$  スーパーカー で、爆速移動 を実現)

---

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。  
手法を厳密に分類してもあまり嬉しいことはないと思いますが、とりあえずこの講習会ではこういう形で整理してみることにします。

# 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:



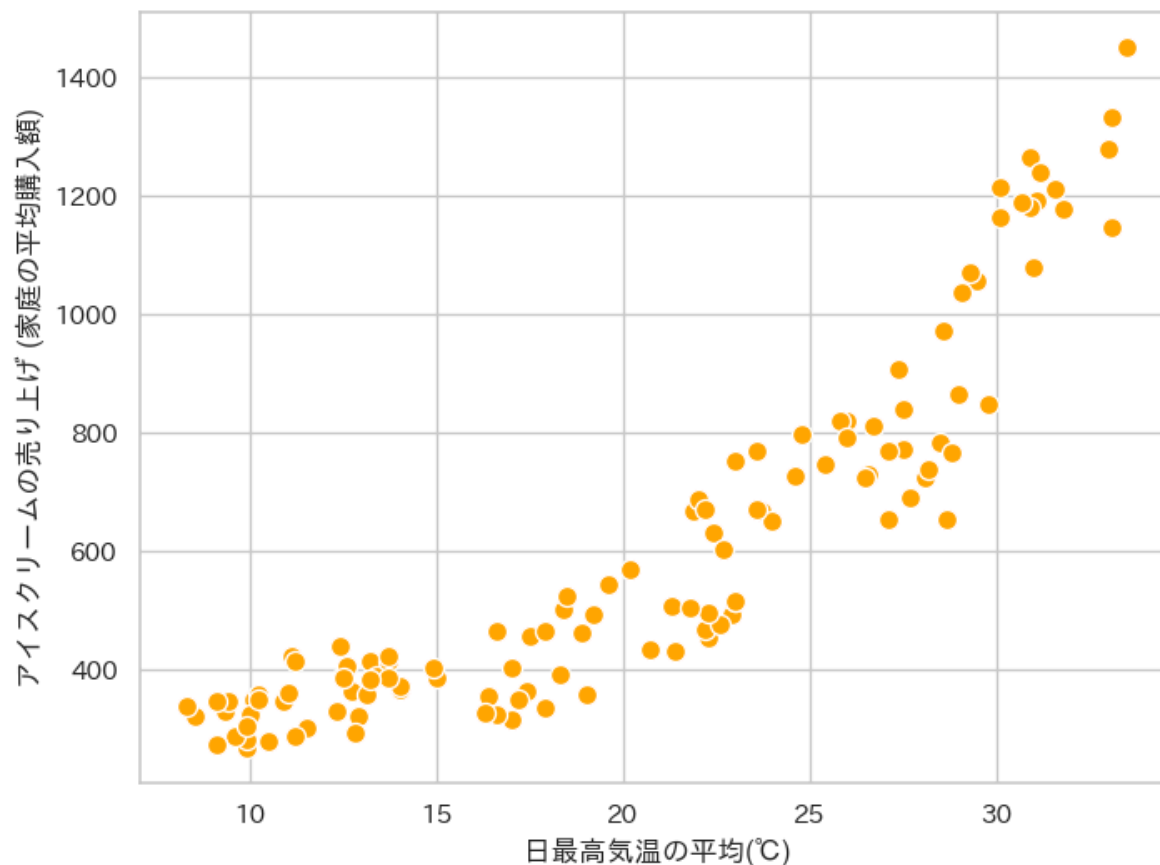
**「学習」を説明できるようになる**

# 「気温」と「アイス」

- 気温↑ → 売れそう
- 気温↓ → 売れなさそう

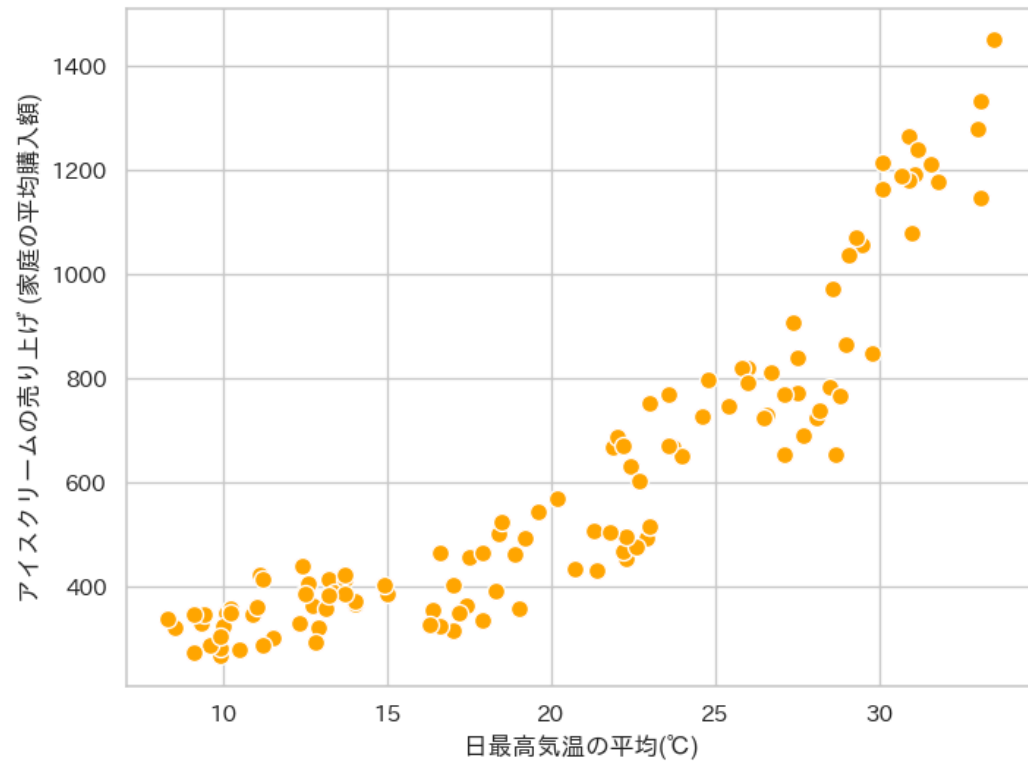
「アイスの売り上げ」は  
「気温」からある程度わかりそう？


🐶 < ...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って嬉しいな。



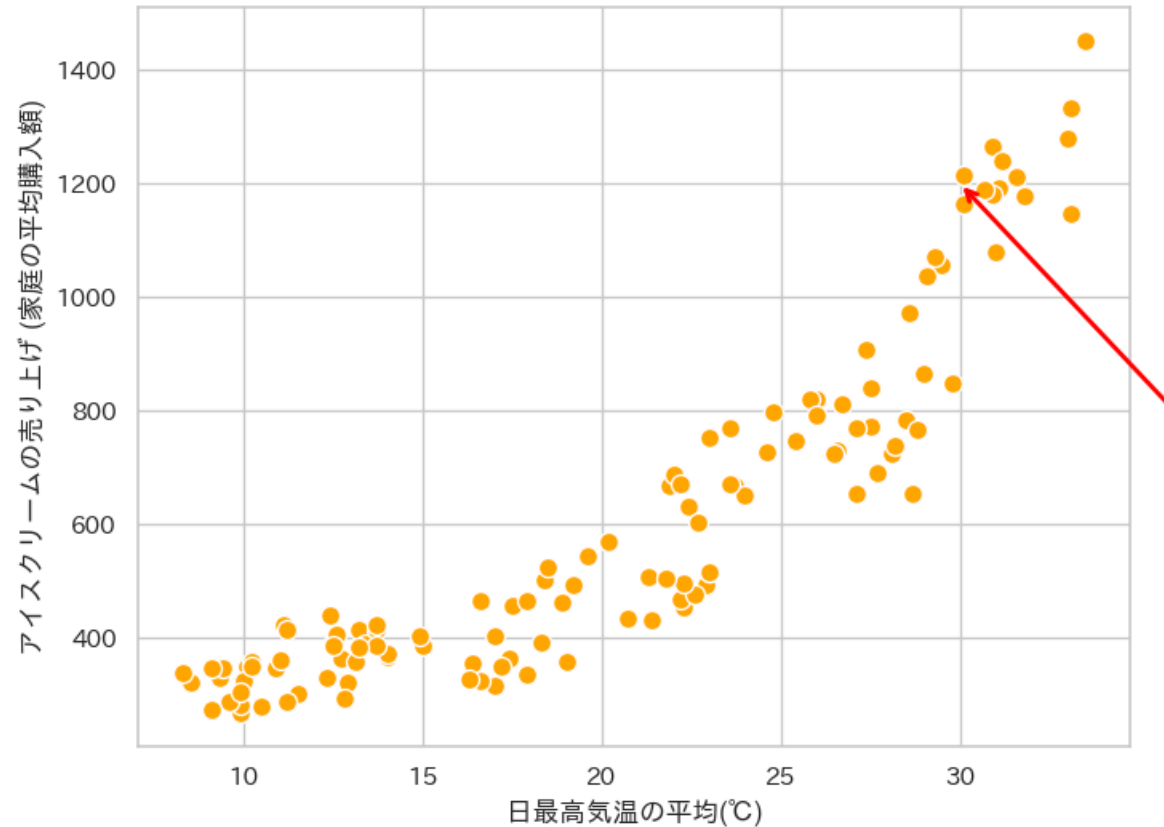
# アイスの売り上げを予測するAIをつくる.

 < 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。



 < !!!!!

# アイスの売り上げを予測するAIをつくる.




< 過去に30°Cのときは...

# 過去を参照すると...

.....

一番簡単な方法: 過去の全く同じ状況を参照する

 < これでアイスの売り上げを予測するAIの完成や！

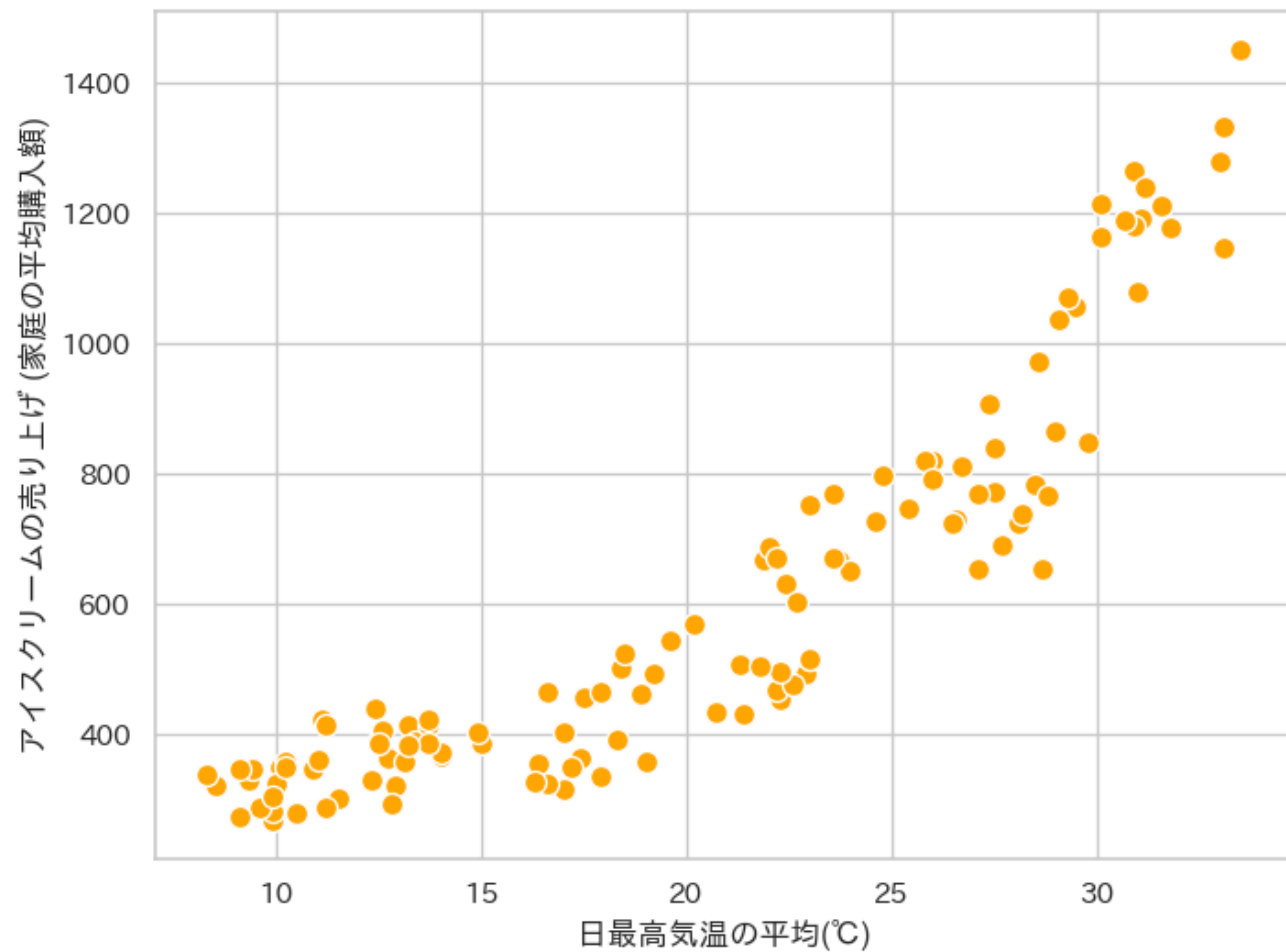


< そのまた来月の予想平均気温は40℃です。

 < !?



# 詰んだ



< 40℃ないやんけ

# 「予測」を考える

.....

「予測」ってなんだっけ？

→ 入力を受け取って、それっぽい出力をすること



今回は、「入力: 気温」 → 「出力: アイスの売り上げ」

そして、 **入力を知ってるものだけとは限らない**

# 予測できるようになる $\leftrightarrow$ ?

---



← 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 =  $f$ (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**と呼ぶ

# 線形回帰

売り上げ =  $f(\text{気温})$  となる関数  $f$  を作りたい。

⇒ 一旦話を簡単にするために

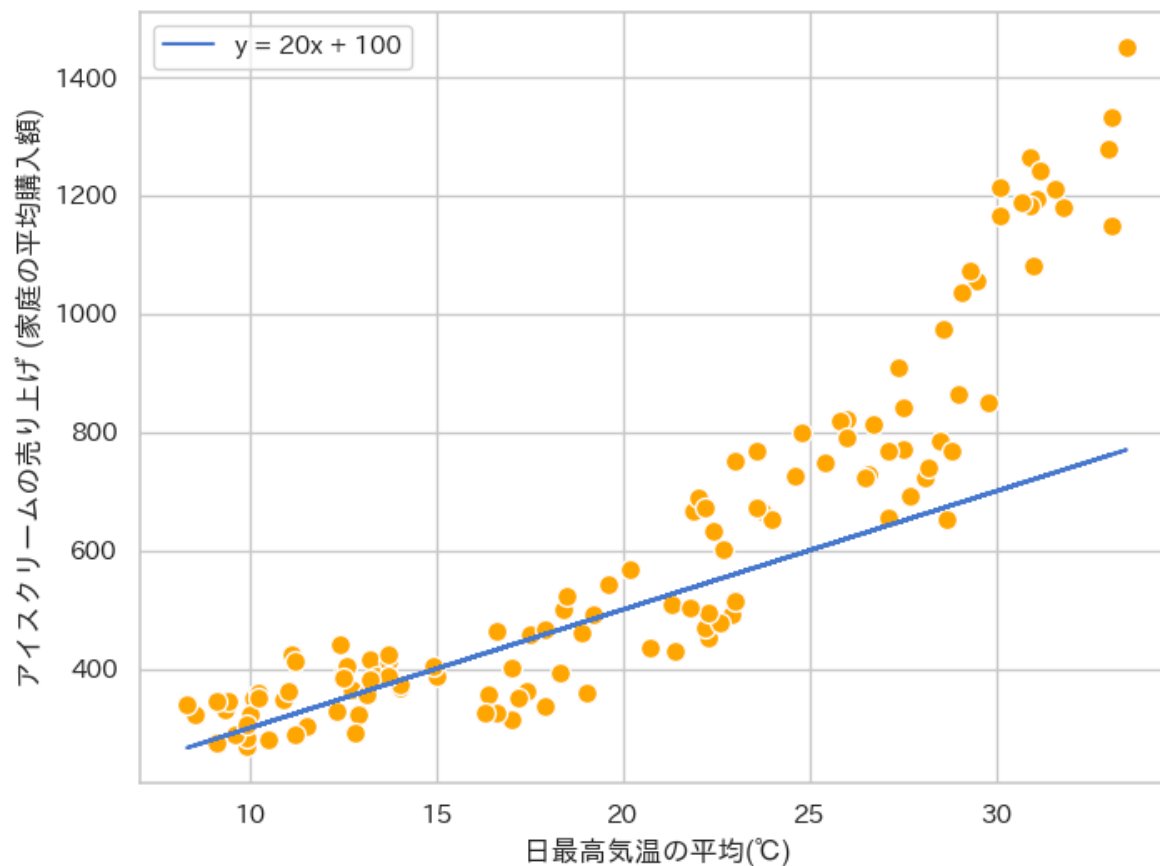
$$\text{「} f(\text{気温}) = a \times \text{気温} + b \text{」}$$

のかたちであることにしてみる。

# ためしてみる

$a = 20, b = 100$  のとき...

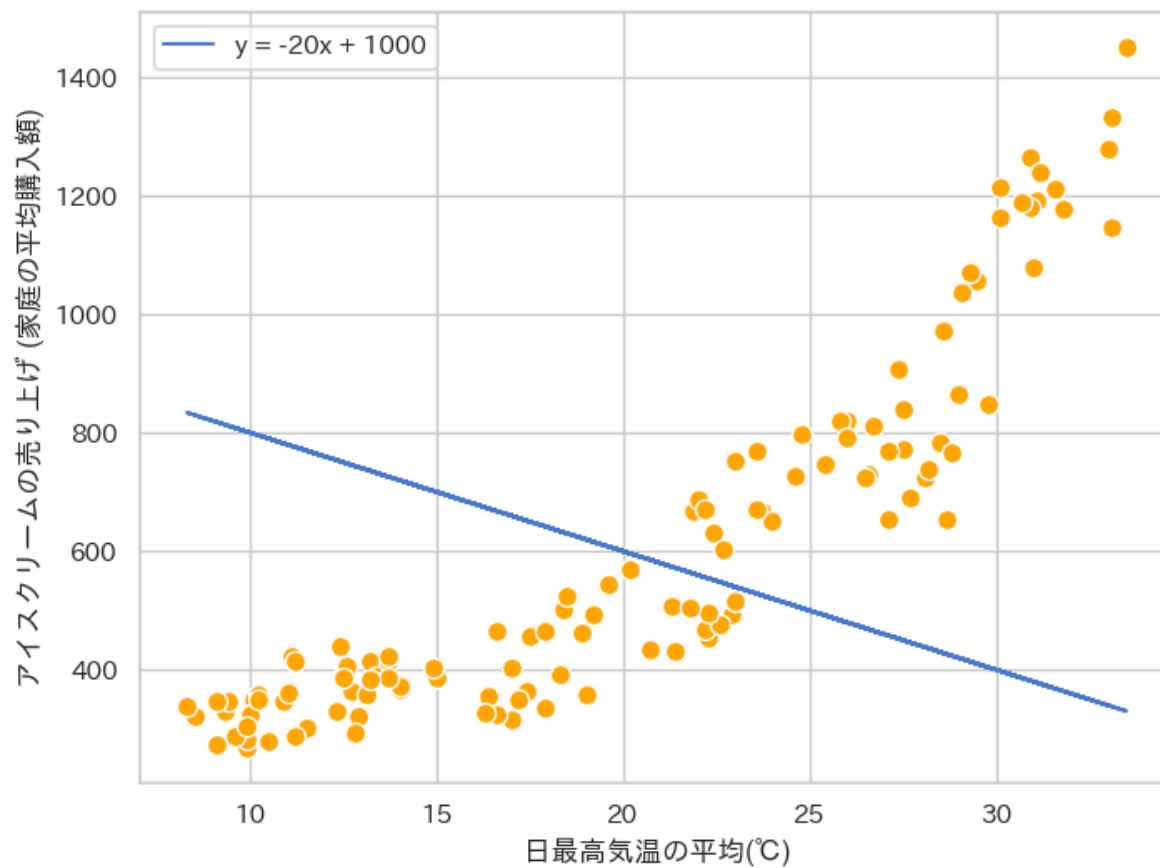
🐻 < わるくない



# ためしてみる

$a = -20$ ,  $b = 1000$  のとき...

🐻 < おわてます



# パラメータ

$a, b$  を変えることでモデル  $f$  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ってモデル自身の性質を定める

● ● ● ● ●  
数を、「**パラメータ**」という。（ $f$  は  $a, b$  をパラメータとして持つ）



$f$  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。
- いったん、今回は関数の形として  $f(x) = ax + b$  (一次関数) に限って、関数を決めることにした。
- 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
- これからやる仕事は、  
「 $a, b$  をいい感じのものにする」ことで「いい感じの  $f$  を作る」こと



# さっきの例

$a = 20, b = 100$  のとき...

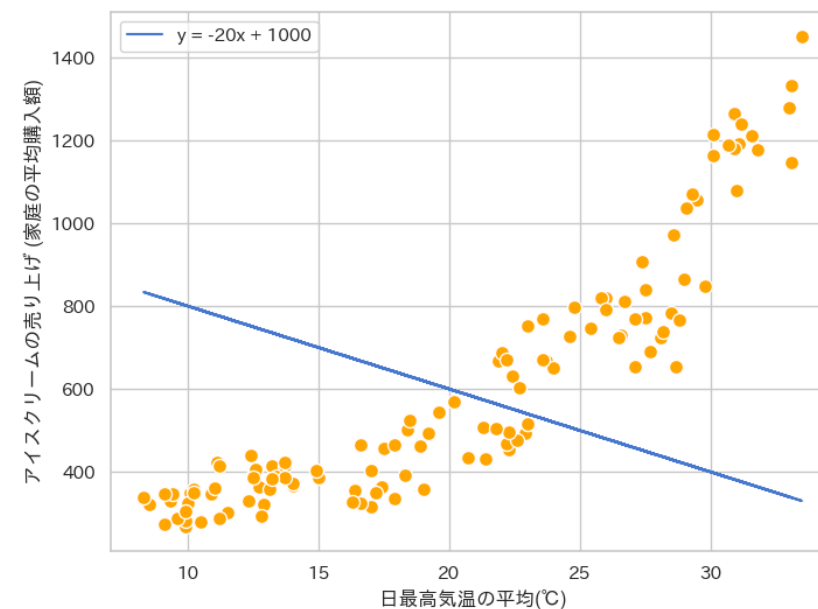
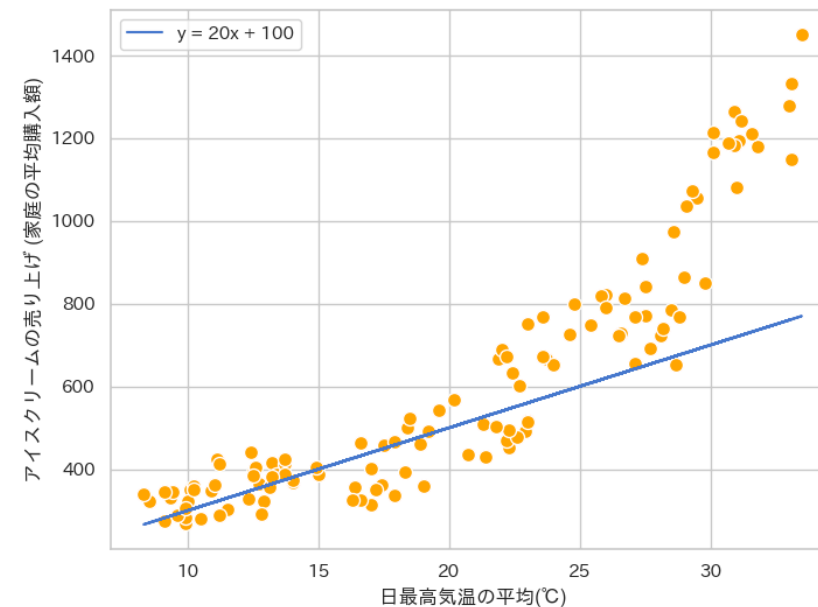
🐻 < わるくない

$a = -20, b = 1000$  のとき...

🐻 < おわてます

↓ なぜ？

🐻 < 見りゃわかる

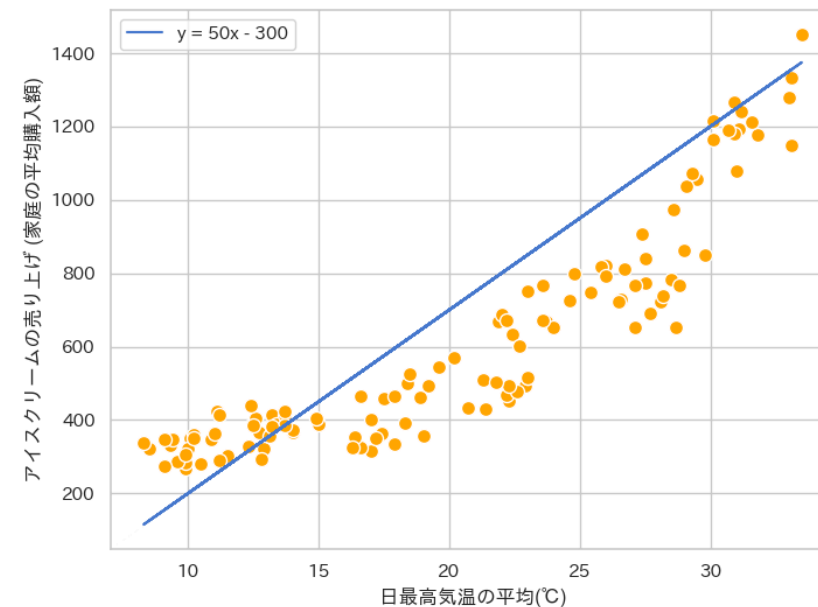
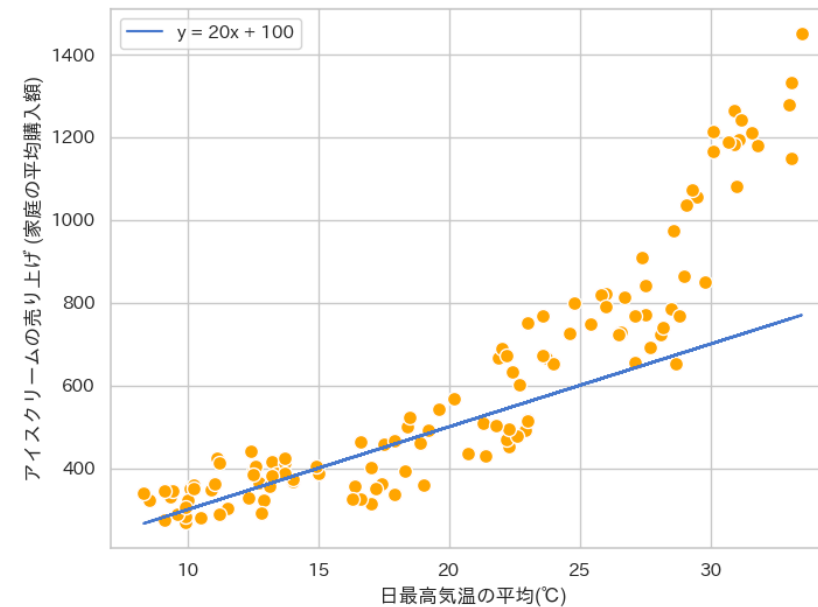


# いい勝負？

.....

上:  $a = 20$ ,  $b = 100$

下:  $a = 50$ ,  $b = -300$



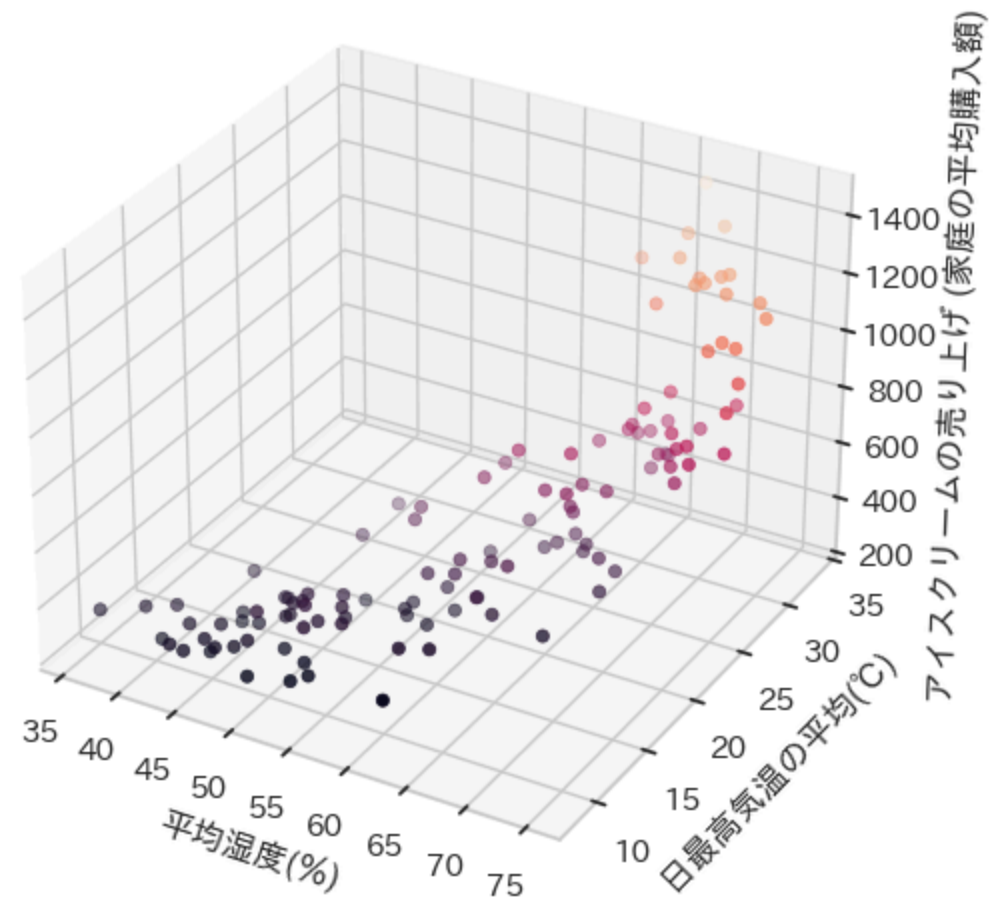
# 破綻

売り上げ =  $f(\text{気温}, \text{湿度}, \dots)$



案1. 高次元の存在になる

案2. 定量的な指標を考える



# 損失関数の導入

良さとは？



悪くなさ



悪くなさとは何か？



データと予測の遠さ

# 平均二乗誤差(Mean Squared Error)

## 平均二乗誤差(Mean Squared Error)

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

$y_i$  : 実際値 (確定値) ... 過去のアイスの売り上げ

$f$  : モデル

$x_i$  : 入力データ (確定値) ... 過去の気温

## 計算例

$\boldsymbol{x} = (50, 80)^T$ ,  $\boldsymbol{y} = (140, 200)^T$ ,  $f(x) = 2x + 50$  のとき、


$$\begin{aligned}\frac{1}{n} \sum_{i=0}^{n-1} (y_i - f(x_i))^2 &= \frac{1}{2} ((140 - (2 \times 50 + 50))^2 + (200 - (2 \times 80 + 50))^2) \\ &= \frac{1}{2} ((140 - 150)^2 + (200 - 210)^2) \\ &= \frac{1}{2} ((-10)^2 + (-10)^2) \\ &= \frac{1}{2} \times 200 \\ &= 100\end{aligned}$$

# 損失関数

.....

このモデルの悪くさを定義する関数を「損失関数」と呼ぶ.

学習とは？

⇒  「損失関数を最小にする  $f$  のパラメータを探す過程」

# 何を動かして損失を小さくする？

Q. 損失は何の関数？（何を動かして、損失を小さくする？）



✓ 各  $x_i, y_i$  は変数みたいな見た目だけど、実際は **「もう観測された確定値」**

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

ものすごく進んだ話: たまに「入力データ」っぽいものに当たるものについても変数とみることもあります。

自分の知っている話だと DeepSDF という三次元形状を表現する NN では latent code と呼ばれる物体固有の表現を表すベクトルも変化させて損失関数を最小化していました。



# いい勝負だったやつの計算例

上:  $a = 20, b = 100$

下:  $a = 50, b = -300$

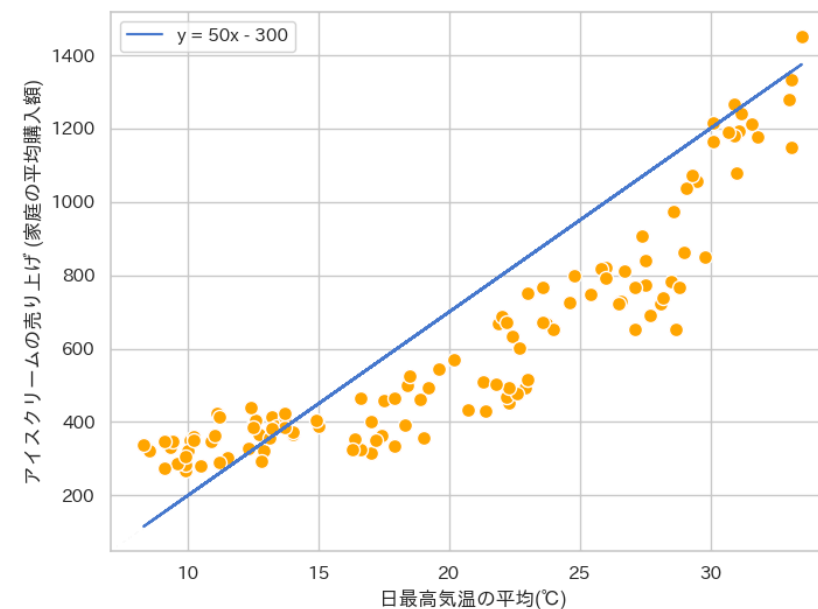
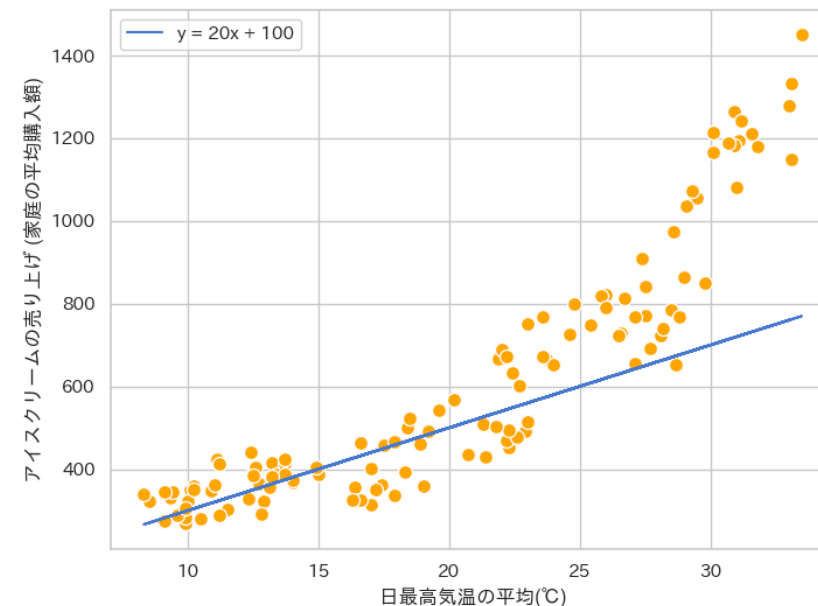
頑張って計算すると、

$$\mathcal{L}(20, 100) = 40268.55$$

$$\mathcal{L}(50, -300) = 39310.45$$



$a = 50, b = -300$  win



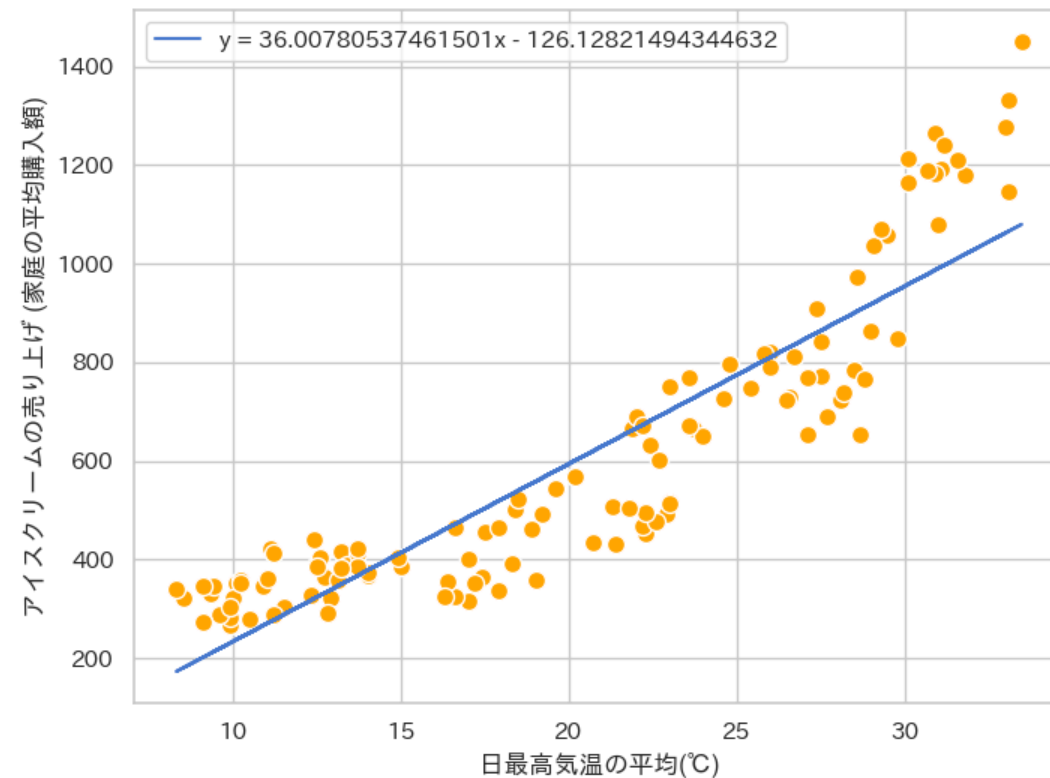
# 実は今回は

.....

$$a = 36.00780537461501$$

$$b = 126.12821494344632$$

で  $\mathcal{L}(a, b)$  が最小



## 当然の疑問

いや  
それ  
どう  
やったの

次回予告

## 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。
- いったん、今回は関数の形として  $f(x) = ax + b$  (一次関数) に限って、関数を決めることにした。
- 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
- パラメータを変えることで損失関数を最小化する過程のことを「学習」と呼ぶ