

やまぐち 高校生データサイエンティスト育成講座



はじめに



本講座概要

全3回となります。
最終的に自由研究を発表いただきます。

#2と#3の間に
メンターとの相談会
を2回実施

第1回 7/20 (月)

AI・データサイエンスの基礎を学ぼう

AIが「魔法の箱」ではなく「データサイエンスに基づくツール」であることを理解してもらう

#	内容
1	オリエンテーション～みなさんがAI・データサイエンスを学ぶ意義～
2	AIとは？
3	【回帰】連続する数値を当てよう
4	【分類】種類を正確に分けよう
5	【ワーク】データ収集してみよう

第2回 8/9 (日)

● データを持ち寄ってAIで分析してみよう

- 自分で集めたデータを使って、AIを動かし、AIを作るためにはどんなデータが必要かの理解を深める

#	内容
1	前回の振り返り
2	持ち寄ったデータを分析してみよう
3	分析結果を発表しよう
4	データ分析の事例紹介
5	【ワーク】自由研究テーマを考えてみよう

第3回 10/18 (日)

● 自分の成果を発表しよう

- ハンズオンセミナー最終日は自身で決めたテーマに関する成果発表を行う

#	内容
1	発表最終準備
2	成果発表会
3	総評、次回コンペ説明
4	講演（SIGNATE社長齊藤の予定）
5	座談会

本日のスケジュール

時間	内容
10 : 00 ~ 10 : 15	開講にあたって
10 : 15 ~ 11 : 00	オリエンテーション
11 : 00 ~ 11 : 10	Teams環境設定（各チーム毎にメンターと会話できる設定）
11 : 10 ~ 12 : 00	チームビルディング「うちのメンターの最強の紹介文を作って！」
12 : 00 ~ 13 : 00	休憩
13 : 00 ~ 13 : 15	SIGNATE Cloud/Google Colabのアクセス確認
13 : 15 ~ 14 : 15	回帰問題に挑戦してみよう
14 : 15 ~ 14 : 25	休憩
14 : 25 ~ 15 : 25	分類問題に挑戦してみよう
15 : 25 ~ 16 : 35	データ収集計画を立てよう
16 : 35 ~ 16 : 50	チーム毎に計画発表
16 : 50 ~ 17 : 00	まとめ・宿題・アンケート

みなさん 講座へようこそ



本講座は AI・データサイエンス を学ぶ！



本講座を終えると

1. AI/データサイエンスの入り口に立てる

本講座を終えると

1. AI/データサイエンスの入り口に立てる
2. データ分析プログラミングが経験できる

本講座を終えると

1. AI/データサイエンスの入り口に立てる
2. データ分析プログラミングが経験できる
3. ちょっとカッコよくなれる



たとえば

RootMeanSquaredError

Jupyter

Cross-Validation

Scikit-learn

Pandas

Python

DeepLearning

lightGBM

Seaborn

Xgboost

LinearRegression

たとえば

RootMeanSquaredError

Jupyter

Cross-Validation

Scikit-learn

だいたいわかるようになります！

DeepLearning

Seaborn

LinearRegression



Boost

これから話すこと

① なぜAI・データサイエンスなのか？

これから話すこと

- ① なぜAI・データサイエンスなのか？
- ② **本講座で心がけていただきたいこと**

なぜAI・データサイエンスなのか？



DX

DXとは？

Digital Transformation (DX)

企業がビジネス環境の激しい変化に対応し、データとデジタル技術を活用して、顧客や社会のニーズを基に、製品やサービス、ビジネスモデルを変革するとともに、業務そのものや、組織、プロセス、企業文化・風土を変革し、競争上の優位性を確立すること。
(出典：経済産業省, DX推進ガイドライン)

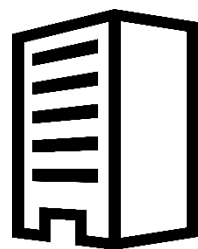
DXの全体像

目的は「競争上の優位性を確保すること」である。

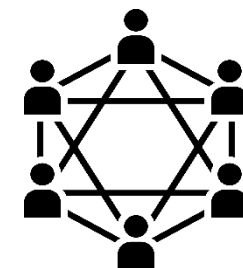
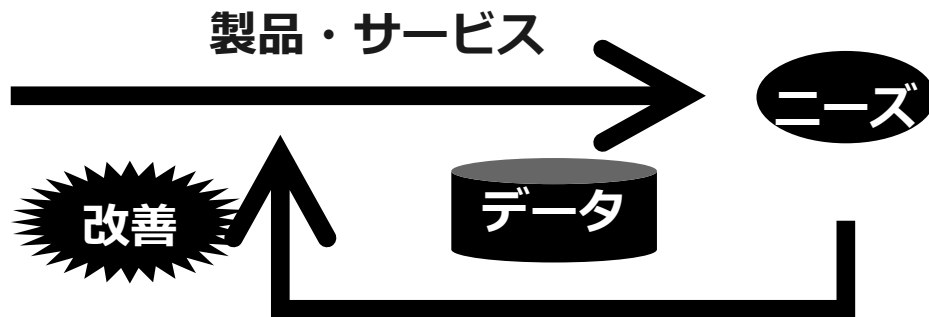
① ビジネスモデル設計

② 製品をデジタルで提供

③ ニーズ計測・データ化



企業



顧客・社会

⑤ ビジネスモデル変革

④ データ・デジタル技術活用

⑥ 企業変革

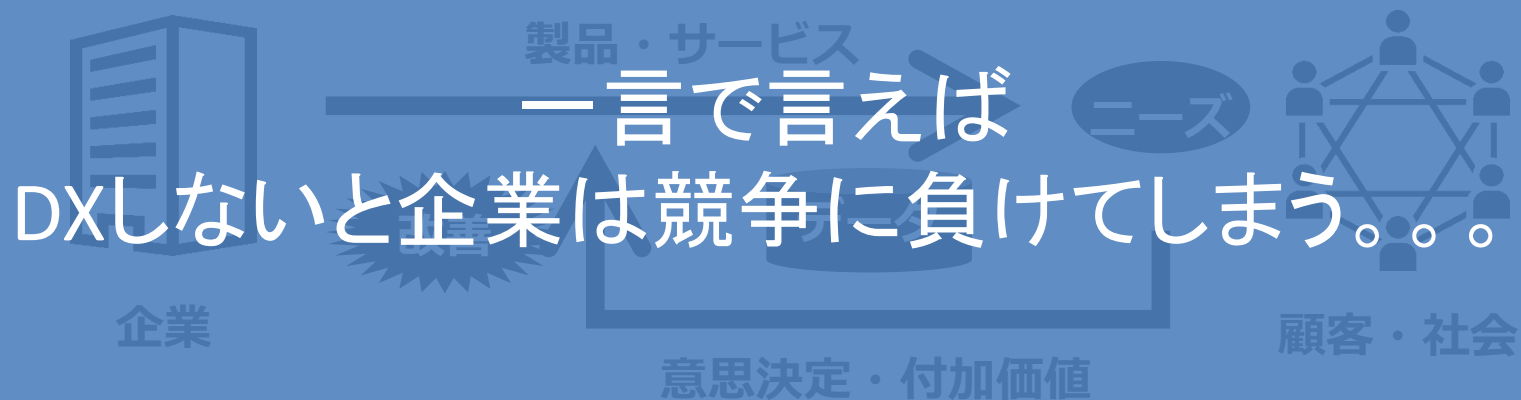
DXの全体像

目的は「競争上の優位性を確保すること」である。

① ビジネスモデル設計

② 製品をデジタルで提供

③ ニーズ計測・データ化



⑤ ビジネスモデル変革

④ データ・デジタル技術活用

⑥ 企業変革

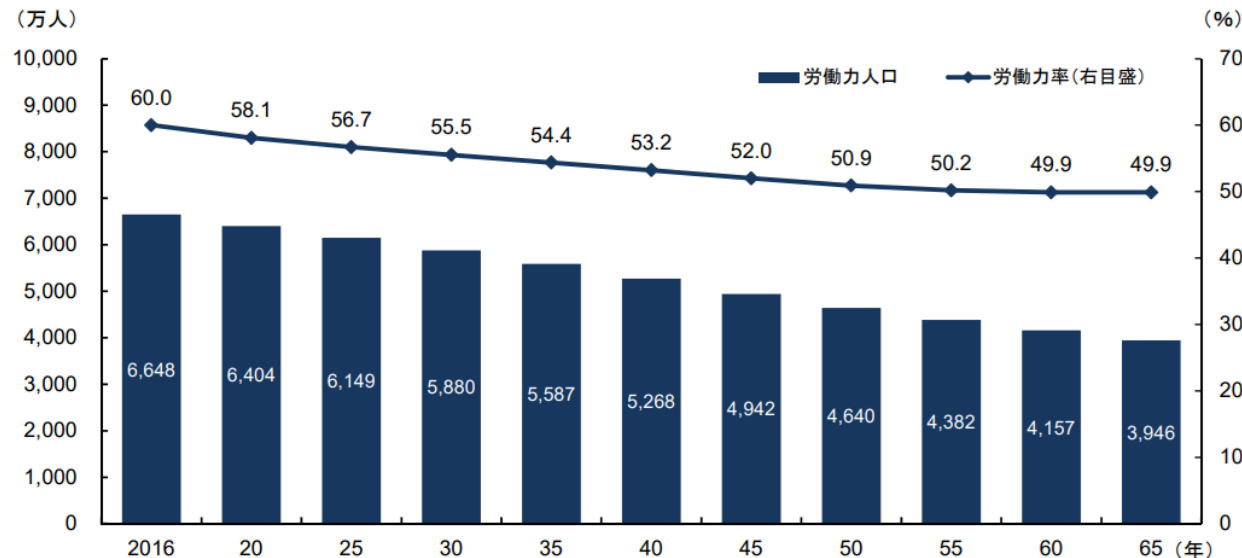
なぜSociety5.0やDXが必要か？

1. 労働力と競争力の低下
2. レガシーシステムの老朽化
3. デジタル技術の波に乗り遅れる

①労働力と競争力の低下

日本はこれから、更なる人手不足が進む。

抜本的な見直しをしないと、人手不足が事業の継続に直結する課題になり得る。



(注) 2016年は実績。2020年以降は、男女別、年齢5歳階級別の労働力率を2016年と同じとして算出（75歳以上は、2016年の75歳以上の労働力率を75～79歳の労働力率とし、80歳以上はゼロとして算出）。

(資料) 総務省「労働力調査年報」（2016年）、国立社会保障・人口問題研究所「日本の将来推計人口」（2017年4月推計）より、みずほ総合研究所作成

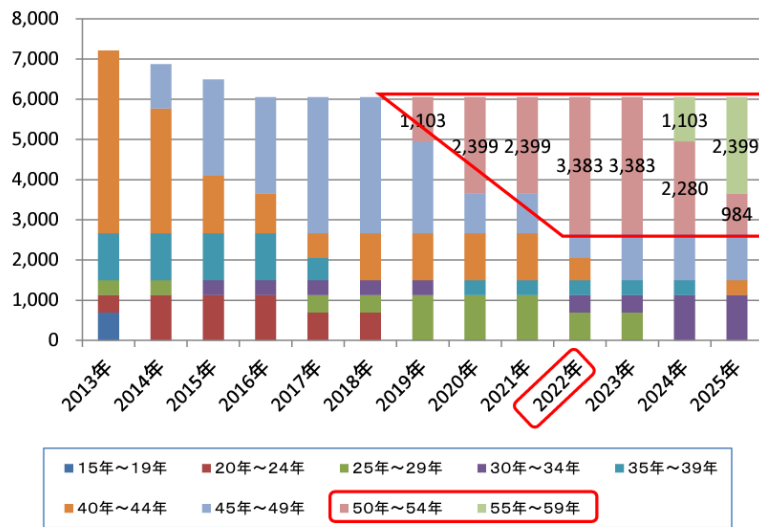
参照: みずほ総合研究所: 少子高齢化で労働力人口は4割減

<https://www.mizuho-rt.co.jp/publication/mhri/research/pdf/insight/pl170531.pdf>

②レガシーシステムの老朽化

高度成長期に建設されたプラント等の産業インフラ設備の老朽化が進行し、修繕費も増加。
更に労働人口が減少にもかかわらず、保守・運用のための人材は変わらないorむしろ増加する傾向。

○2022年には、半数以上が稼働年数50年以上が半数以上となる

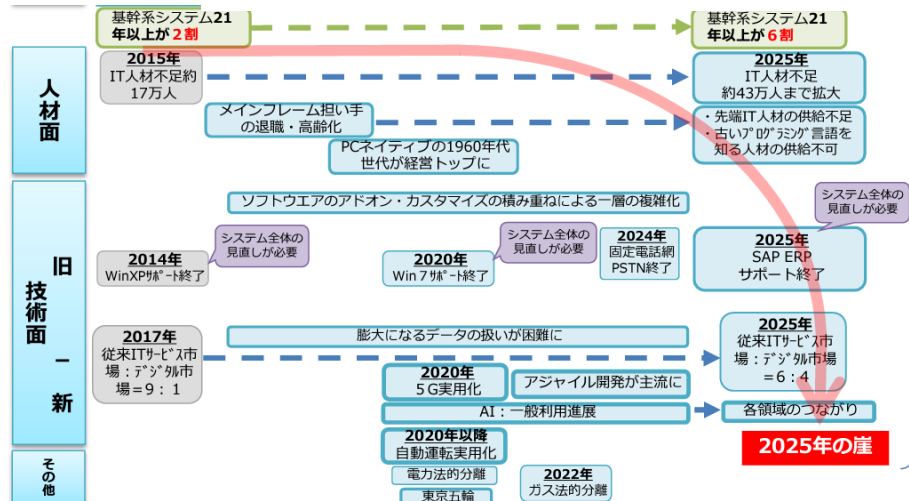


(資料) 日本の石油化学工業50年データ集(重化学工業通信社)

引用:「今後の産業保安行政のあり方について」

https://www.meti.go.jp/shingikai/sankoshin/hoan_shohi/pdf/004_03_01.pdf

引用:「DXレポート～ITシステム「2025年の崖」の克服とDXの本格的な展開～」
https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/pdf/20180907_01.pdf



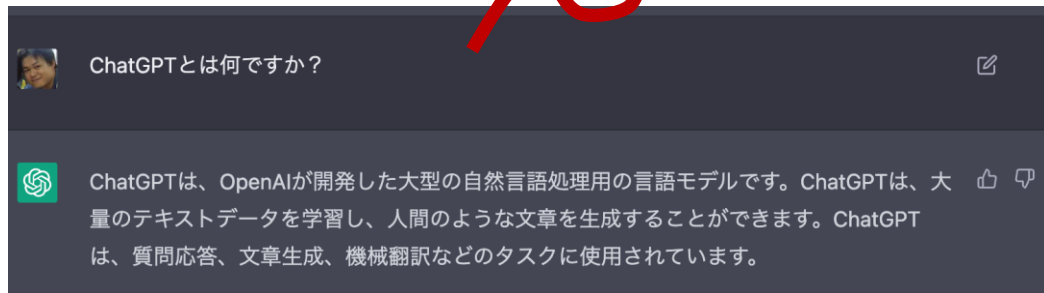
これはDXレポートで言われる
「2025年の崖」と同様

③ デジタルの波に乗り遅れる

AIの進化により、状況がどんどん変わり始めている。

ChatGPT

プロンプト



約100億ドル出資



③デジタルの波に乗り遅れる

生成AIと協働することで、1人で何人分もの価値を生み出す可能性を秘めている

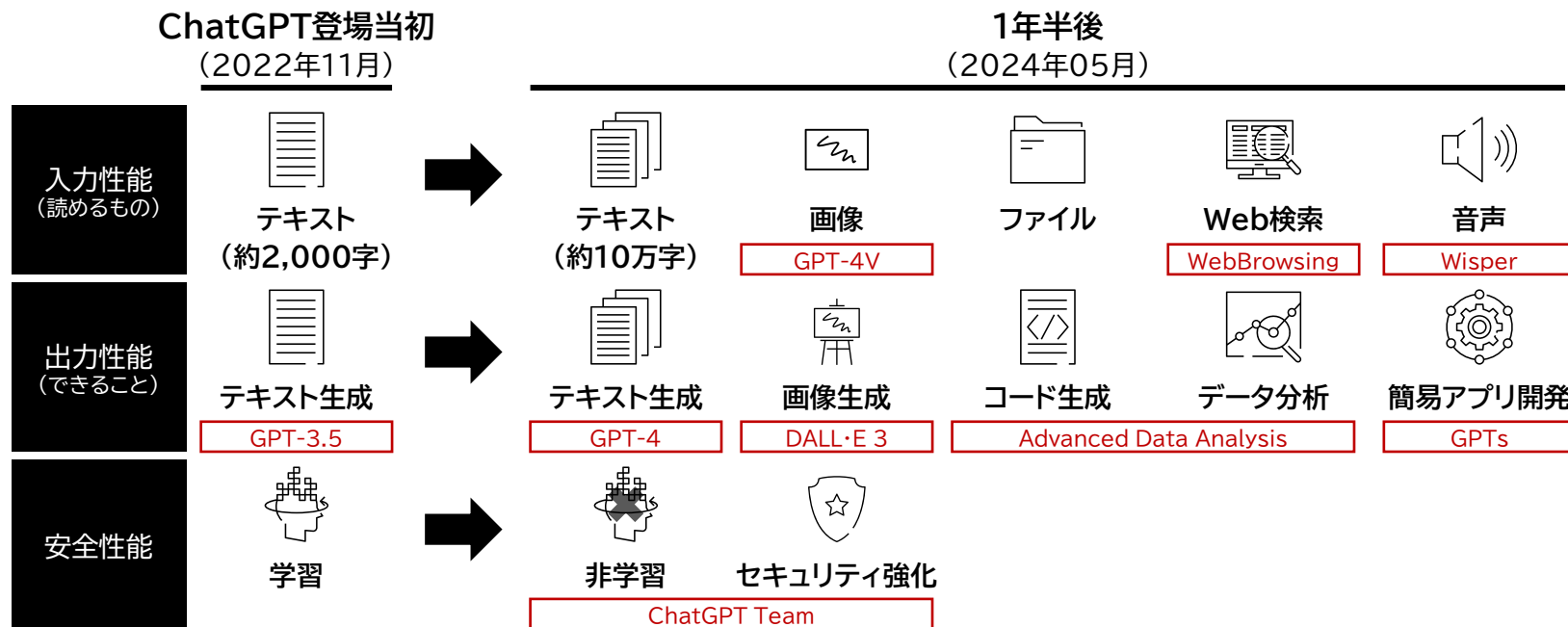
□現在はレベル2→3への過渡期

レベル	名称	説明
1	Chatbots	会話能力を持つ
2	Reasoners(推論者)	人間レベルの問題解決能力を持つ
3	Agents(エージェント)	自律的に意思決定や行動ができる
4	Innovators	発明や発見ができる
5	Organizations	組織全体の仕事ができる

※Bloomberg, (2024/7/11), OpenAI Scale Ranks Progress Toward 'Human-Level' Problem Solving.
<https://www.bloomberg.com/news/articles/2024-07-11/openai-sets-levels-to-track-progress-toward-superintelligent-ai>

③ デジタルの波に乗り遅れる

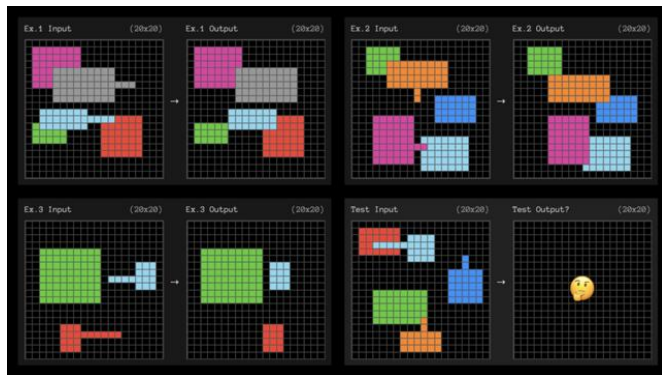
- レベル1:チャットボット(2023-2024)
- 「会話」能力が急速に進化(マルチモーダル化)



③ デジタルの波に乗り遅れる

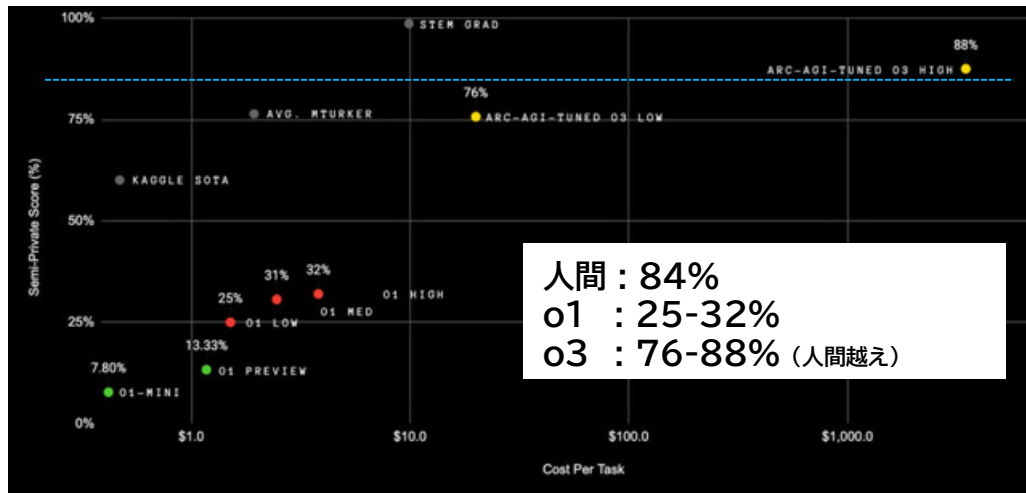
- レベル2: リーズナー/推論者 (2025-)
- 生成AIの知能が、いよいよ「人間越え」
 - 推論モデル(o1/o3)の登場。複雑なタスクに対して、多くの時間をかけて推論を行う
 - 科学、数学、コーディングなどで優れた性能、博士のトップレベル水準
 - AIが苦手な抽象度の高い推論でも、人間越え

AGIの性能を評価するベンチマーク
「ARC-AGI」



Source : <https://www.techno-edge.net/article/2024/12/23/3949.html>

抽象度の高い推論でも、人間越えを実現



人間 : 84%
o1 : 25-32%
o3 : 76-88% (人間越え)

③ デジタルの波に乗り遅れる

- レベル3: エージェント
- 自律的×実行を伴う→複雑な作業を丸っとお任せ！



“1人ユニコーン企業”
たった1人で10億ドルの
価値がある会社を作れる？



社長



社員(AIエージェント)

③ デジタルの波に乗り遅れる

- レベル3: エージェント
- 自律的×実行を伴う→複雑な作業を丸っとお任せ！



何もしないとやばそう...

“1人ユニコーン企業”

たった1人で10億ドルの価値がある会社を作れる？



社長

社員(AIエージェント)

ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

「機能的価値」＝品質のよいもの

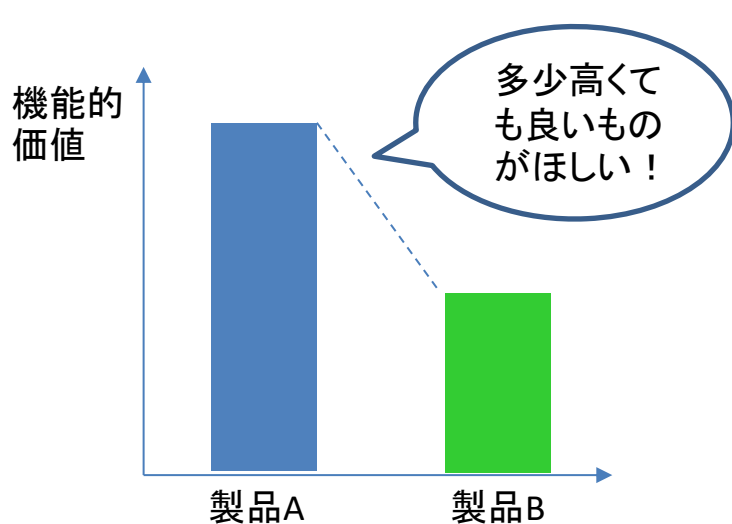
「意味的価値」＝持ってる／使ってることがイケてる！と思える

ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

「機能的価値」＝品質のよいもの

「意味的価値」＝持ってる／使ってることがイケてる！と思える



高田の携帯電話の歴史



2000

モノクロ



2001

画面カラー
+デジカメ



2002

6.4和音
着うた



2005

カメラ良くなる
着うたフル



2007

画面大きくなる
カメラ高機能



2009



2011

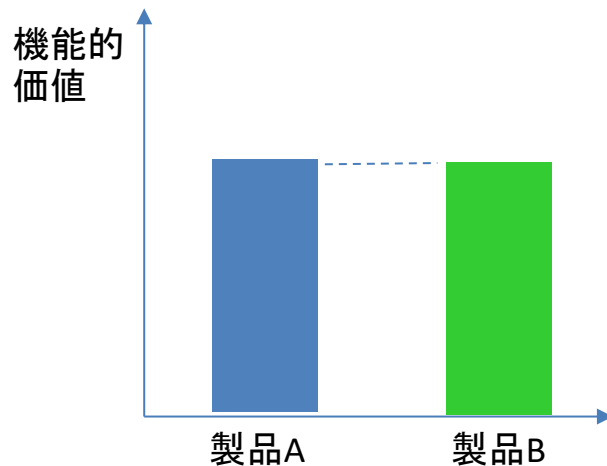
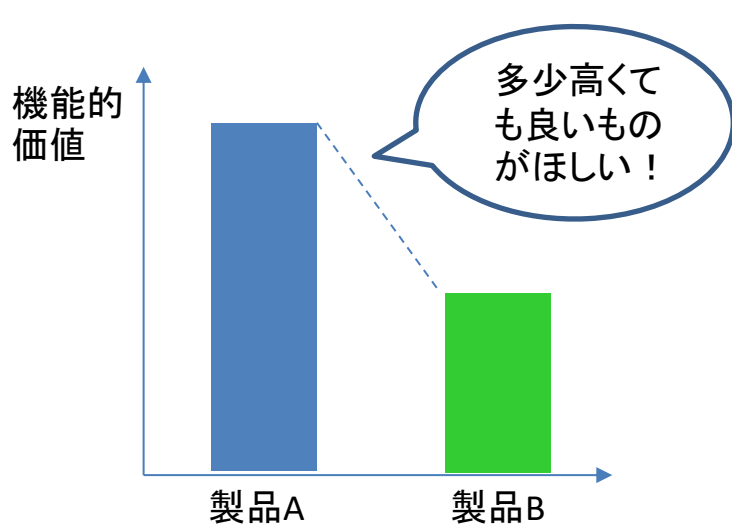
スマホになる
3G→4G、容量大

ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

「機能的価値」＝品質のよいもの

「意味的価値」＝持ってる／使ってることがイケてる！と思える



ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

「機能的価値」＝品質のよいもの

「意味的価値」＝持ってる／使ってることがイケてる！と思える

機能的
価値

製品A

製品B

多少高くても
良いものがほしい！

機能的
価値

製品A

製品B

安いほうがいい！
デザインが良いのが良い！
やりたいことができる方がいい！

ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

①安く早く作れる方が強い

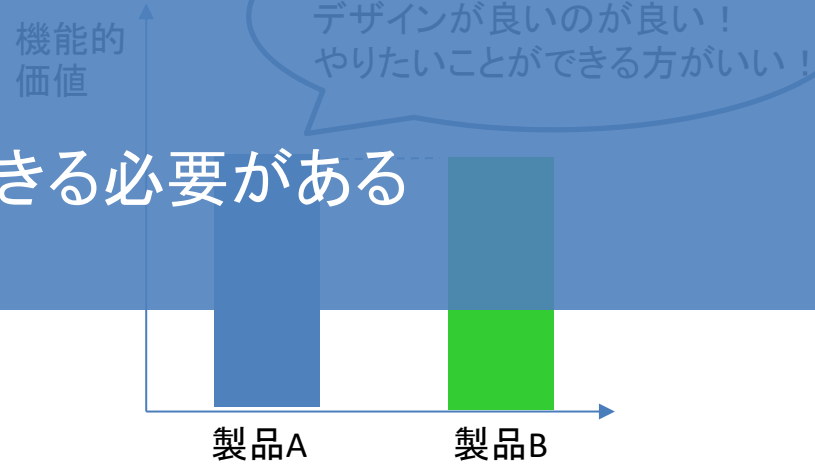
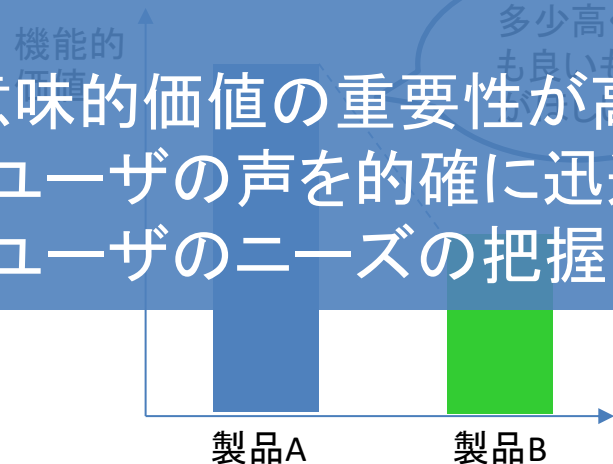
⇒製造工程をより効率化しなければ実現できない

⇒省人化、自動化

②意味的価値の重要性が高くなった

⇒ユーザの声を的確に迅速に対応できる必要がある

⇒ユーザのニーズの把握



ものづくりの観点からみると？

商品やサービスが顧客に与える価値＝「機能的価値」＋「意味的価値」

①安く早く作れる方が強い

⇒製造工程をより効率化しなければ実現できない

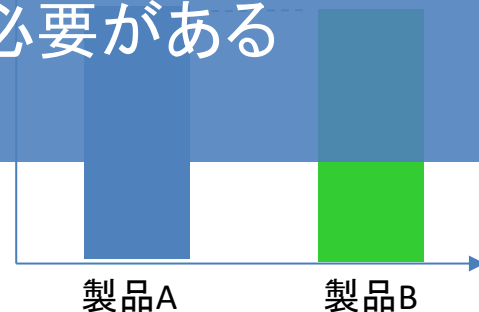
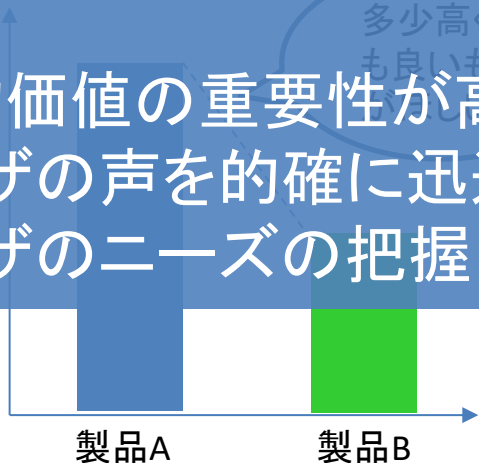
⇒省人化、自動化

②意味的価値の重要性が高くなった

⇒ユーザの声を的確に迅速に対応できる必要がある

⇒ユーザのニーズの把握

AI・データ分析のスキル
が確実に武器になる時代



②AIってなんだろう？



AIって聞くと何をイメージしますか？

生成AIに聞いてみました



なんでこの画像にしたの？

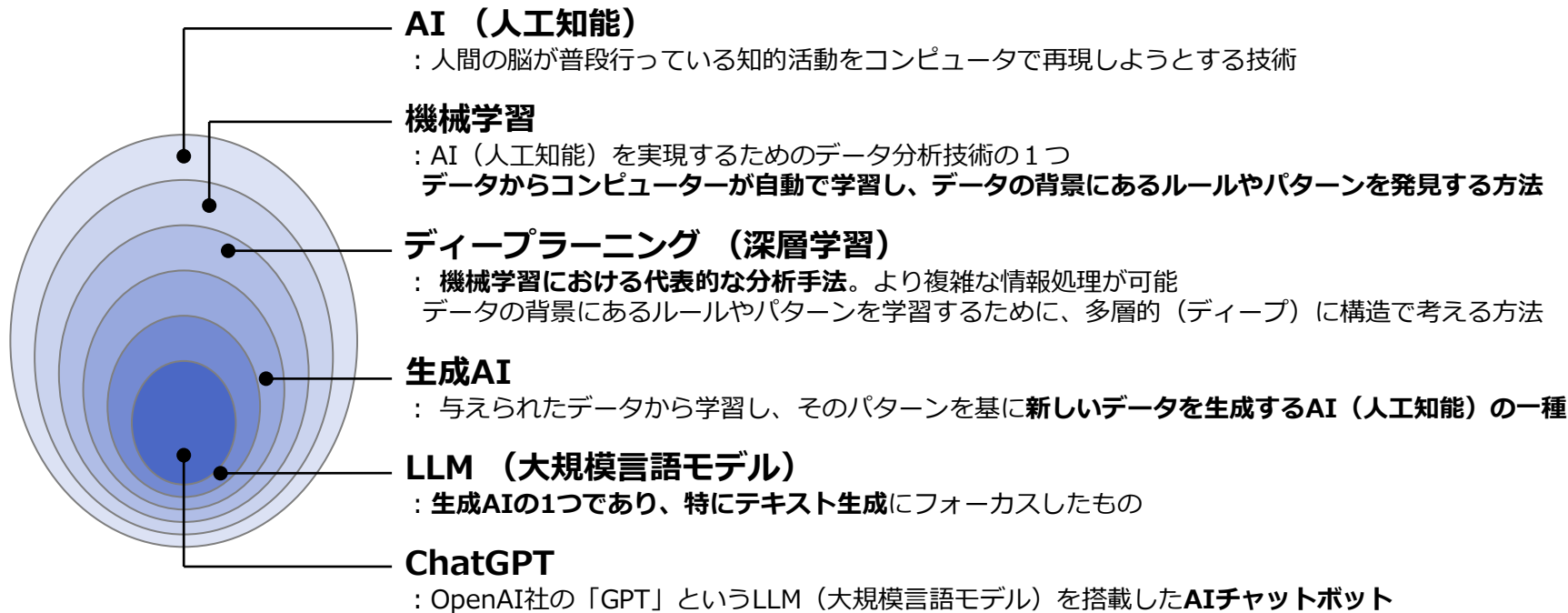
この画像にしたのは、**AIを「人間の代わりに考えるロボット」だけではなく、いろいろな技術が組み合わさった仕組みとして表したかったから**です。中央の光る脳は、AIが大量のデータをもとに、特徴やパターンを学ぶ様子を表しています。周囲のグラフや数字は、AIがデータを分析して、予測や判断を行うことを示しています。

また、画像にはロボットや機械だけでなく、人間も描かれています。これは、AIが人間を完全に置き換えるものではなく、**人間と一緒に仕事をしたり、考えることを助けたりする存在**だからです。

つまり、この画像ではAIを、

「データから学び、分析や創作を行い、人間を助ける技術」
として表現しています。

AIの概念整理



機械学習の仕組み

- データから“パターン”を学ぶ
-

[3、 4]	=	1 1
[6、 5]	=	2 9
[4、 9]	=	3 5
[5、 2]	=	9
[3、 8]	=	2 3



どんな法則？

機械学習の仕組み

- データから“パターン”を学ぶ
-

$$\begin{array}{lcl} [3、4] & = & 1\ 1 \\ [6、5] & = & 2\ 9 \\ [4、9] & = & 3\ 5 \\ [5、2] & = & 9 \\ [3、8] & = & 2\ 3 \end{array}$$



どんな法則？

$$[X、Y] = \underline{X \text{ かける } Y \text{ ひく } 1}$$

機械学習の仕組み

- データから“パターン”を学ぶ

$$\begin{array}{lcl} [3, 4] & = & 11 \\ [6, 5] & = & 29 \\ [4, 9] & = & 35 \\ [5, 2] & = & 9 \\ [3, 8] & = & 23 \end{array}$$

じゃあこれは？

$$[4, 5] = ?$$



どんな法則？

$$[X, Y] = \underline{X \text{ かける } Y \text{ ひく } 1}$$

機械学習の仕組み

- データから“パターン”を学ぶ

$$\begin{array}{lcl} [3, 4] & = & 11 \\ [6, 5] & = & 29 \\ [4, 9] & = & 35 \\ [5, 2] & = & 9 \\ [3, 8] & = & 23 \end{array}$$



どんな法則？

$$[X, Y] = \underline{X \text{ かける } Y \text{ ひく } 1}$$

じゃあこれは？

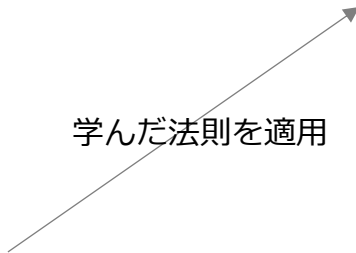
$$[4, 5] = ?$$



19

予測できた！

学んだ法則を適用



AIが得意なこと

分類

- 写真を見て犬か猫かを分類
- たけのこ派かきのこ派かを分類



推薦

- 好みに合いそうな動画や音楽を推薦



予測

- 商品の売れ行きを予測
- 来場客数を予測



推薦

- 指示に合わせ文章や画像を生成



AIにも苦手なことがある

AIは学習したパターンには強い一方、初めての状況や例外的な判断は苦手です。

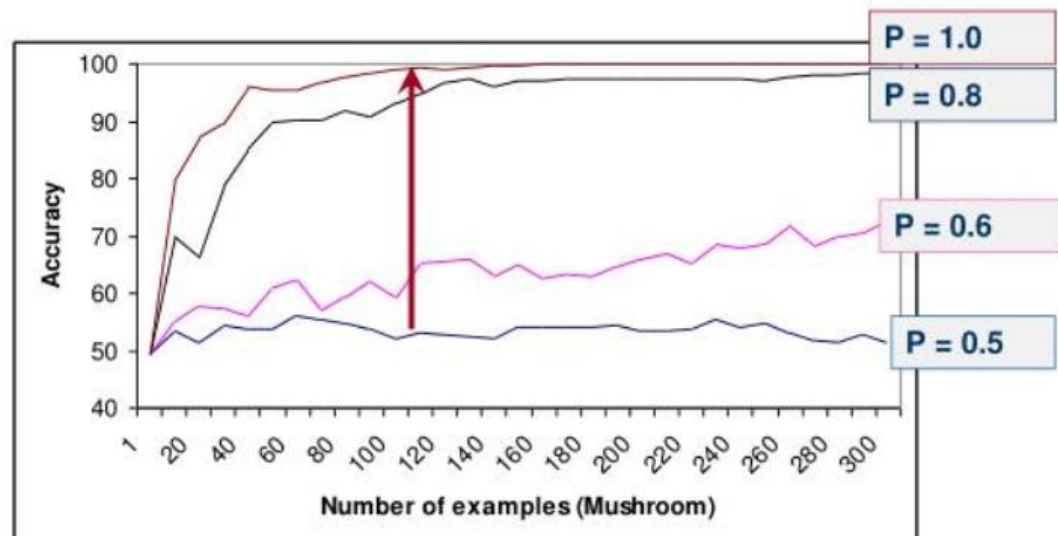
予測や分類を行うAIは、過去のデータに似た状況ほど正確に判断できます。一方、学習していないケースや、背景を考える必要がある場面では間違えることがあります。

AIが苦手になりやすいこと	高校生にも身近な例	なぜ苦手なのか
初めて見るものを判断する	犬と猫だけを学習したAIに、キツネの写真を見せる	学習した分類の中から無理に答えを選ぶようになるため
いつもと違う状況に対応する	晴れの日の写真で学習した顔認証が、暗い場所やマスク姿ではうまく認識できない	学習時と実際の条件が異なるため
少ない事例や珍しいケースを予測する	過去にほとんど例がない大雨や災害の被害を予測する	判断材料となるデータが少ないため
数字だけでは分からない事情を考える	テスト結果だけを見て、「努力していない生徒」と判断する	本人の体調や家庭環境など、データにない背景を理解できないため
偏ったデータから公平に判断する	特定の年代や地域のデータだけで、全員の傾向を予測する	学習データの偏りが、結果にも表れるため

AIの賢さは「データ」で決まる

Quality and Classification Performance

Labeling quality increases → classification quality increases



機械学習ではデータ量は重要！
一方、単にデータを集めても十分な精度が出ず、質も担保しなければならない。

AIと人間の役割

AIを作る人・使う人はAIのことを理解の上で「正しく」使うことが重要

AIに任せやすいこと	人間が担うこと
大量のデータを早く処理する	何を知りたいのか、目的や問いを決める
データの中から傾向やパターンを見つける	見つけた傾向の意味や理由を考える
過去のデータを元に予測・分類する	予測結果が妥当か、現実と照らして確認する
条件に合う候補や案を複数出す	目的に合う案を選び、よりよく修正する
決められた作業を繰り返し行う	例外や想定外の状況に対応する
データを元に判断を支援する	最終的な判断を行い、その結果に責任を持つ

AIは判断を助ける道具。最終的に考え、決めるのは人間。

③本講座で心がけて頂きたいこと



これから勉強することには、、、

これから勉強することには、、、

基本

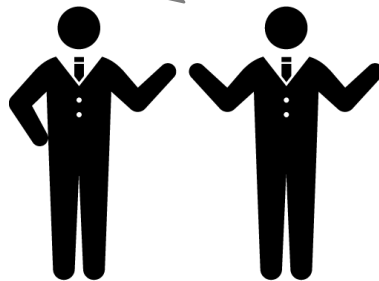
正解はありません



たとえば…

プログラミング の書き方

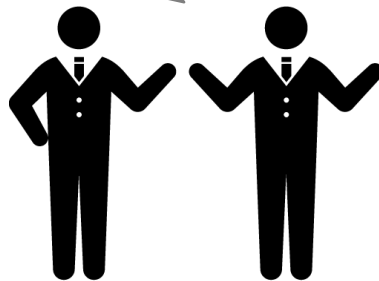
こっちの方が見やすい！
こっちが動作速くない？



たとえば…

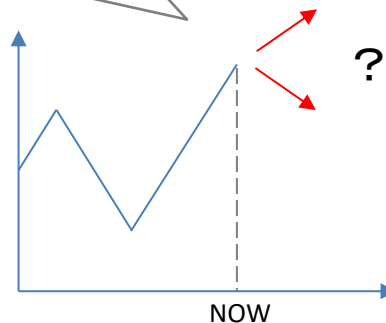
プログラミング の書き方

こっちの方が見やすい！
こっちが動作速くない？



データの解釈

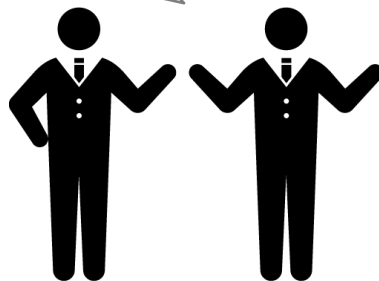
周期的に次は下がる！
いやそのまま上昇する！



たとえば…

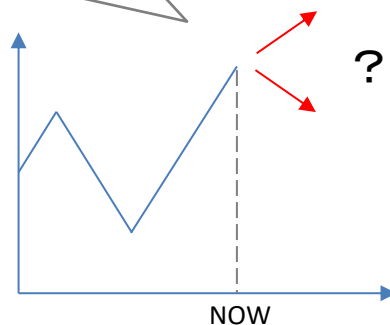
プログラミング の書き方

こっちの方が見やすい！
こっちが動作速くない？



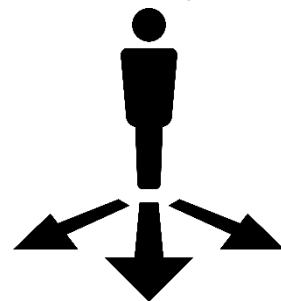
データの解釈

周期的に次は下がる！
いやそのまま上昇する！



アプローチ方法

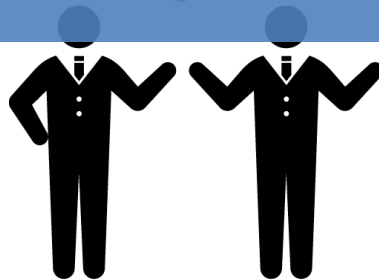
右が正解？左が正解？
いや真ん中もありかも…



たとえば…

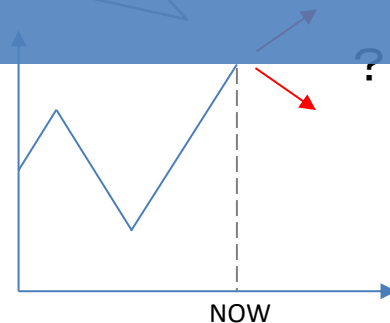
プログラミング の書き方

こっちの方が見やすい！
こっちが動作速くない？



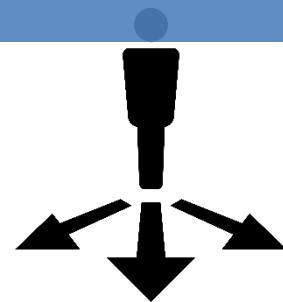
データの解釈

定期的には次は下がる！
いやそのまま上昇する！



アプローチ方法

右が正解？左が正解？
いや真ん中もありかも…



じゃあどうすればいいの？

心構え 3 か条

1. 試行錯誤をする・調べる

- やって見ないとわからない
- 調査する力（ググる力・生成AI活用）は本当に重要

心構え 3 か条

1. 試行錯誤をする・調べる

- やって見ないとわからない
- 調査する力（ググる力・生成AI活用）は本当に重要

2. 自分なりの意見／仮説を持つ・議論する

- 考えの整理・洗練化

心構え 3 か条

1. 試行錯誤をする・調べる

- やってみたいとわからない
- 調査する力（ググる力・生成AI活用）は本当に重要

2. 自分なりの意見／仮説を持つ・議論する

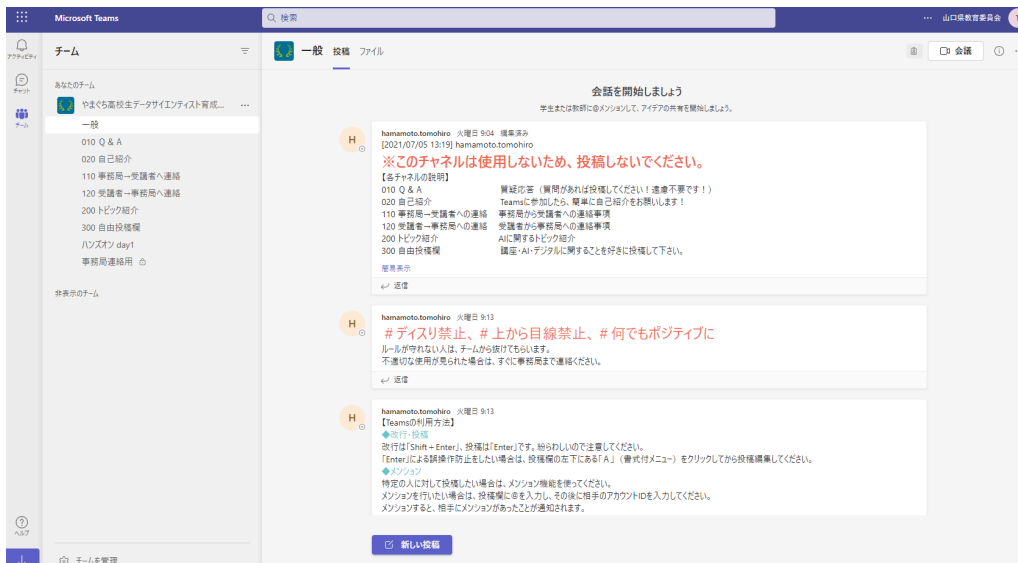
- 考えの整理・洗練化

3. 質問する・教え合う

- “わからない”をそのままにしない
- 教えることができるようになる近道



もっと聞きたいこと・知りたいことがあれば Teamsで質問をお待ちしています！



さあデータサイエンスを始めましょう



応援してます！

チームビルディング



Teams環境の準備

- チーム毎にメンターと会話できるようにします
- チーム毎にメンターと話せる環境となっているかを確認しましょう
- 皆さんもTeamsも起動し、本講座のチャンネルが開けるかどうかを確認しましょう
- 更にTeamsのチャット機能を使い、自身のTeamsにて投稿ができるかどうかも確認しましょう（自由投稿欄チャンネルを使いましょう）

これからやること

お題

「うちのメンターの最強の紹介文を作ってください」

これから皆さんは、Day1～Day3を通じてメンターに相談しながらAI・データ分析に取り組みます。そこで、まずは自分たちのメンターがどんな人で、どんな時に頼るとよいのかを取材し、他のチームに紹介してください。

手順	時間	内容
①	11:15 - 11:25	メンター・メンバーへの自己紹介
②	11:25 - 11:35	メンターへの質問を考える
③	11:35 - 11:45	メンターへインタビュー
④	11:45 - 11:55	インタビューした内容を整理し、Teamsに投稿
⑤	11:55 - 12:00	チーム毎にメンターを紹介

①メンター・メンバーへの自己紹介

まずチーム内でメンター・メンバーに向け自己紹介をしましょう

自己紹介の例

- 名前、学校、学年
- 好きな教科、部活動、趣味など
- 得意なこと、自分の特徴（アイデアをたくさん出すタイプ、じっくり考えるタイプ等）
- 講座への参加理由、データ分析やプログラミング経験など
- 講座でできるようになりたいこと、サポートしてほしいこと



②メンターへの質問事項を考える（1）

ねらい

- うちのメンターの最強の紹介文を作ること
- メンターにインタビューし、メンターの人柄や得意なこと、相談の仕方が伝わる自己紹介文を作りましょう
- 「最強」とは経歴がすごい！ではなく、困ったときに気軽に話しかけたいと思える自己紹介文が最強です

完成する自己紹介文のイメージ（下記のような要素を入れられるような質問を考えましょう、以下に加えて楽しい要素を入れていただいても構いません）

項目	内容例
基本情報	名前、仕事・勉強の中身
得意分野	詳しいこと、相談できること
高校時代	好きな教科、部活動、当時の悩み
経験	失敗したこと、挑戦したこと
人柄	大切にしていること、考え方

項目	内容例
好きなこと	趣味、好きな食べ物
意外な一面	見た目からは分からない特技経験
相談の仕方	どのように声をかけるとよいか
メンターとしての姿勢	高校生をどのようにサポートした いか

②メンターへの質問事項を考える（2）

考えた質問やメモはteamsにどんどん投稿しましょう

ご自身の所属する
チームの「チャンネル」を探し、その
チャンネル内に投稿
しましょう

The screenshot shows the Microsoft Teams web interface. On the left sidebar, the 'チーム' (Team) section is highlighted with an orange callout box labeled 'チーム'. Below it, the 'チャンネル' (Channels) list is shown, with an orange callout box labeled 'チャンネル' pointing to it. A blue line points from the text on the left to this list. The main area shows a channel conversation. An orange callout box labeled '投稿メッセージ' (Posted Message) points to a message from hamamoto.tomohiro. At the bottom, a blue button labeled '新しい投稿' (New Post) is highlighted with an orange callout box labeled '新しい投稿'.

チーム

チャンネル

投稿メッセージ

新しい投稿

③メンターにインタビューする

メンターにインタビューをしましょう

インタビューで聞いた内容はTeamsにメモとして投稿しましょう

インタビューのコツ

以下のような順序でインタビューするとスムーズかもしれません。

質問は、一問聞いて終わりではありません。
回答の中に気になる言葉があったら、もう
一歩聞いてみましょう。

- ① 話しやすい質問
 - ・ 普段はどのような仕事・勉強をしていますか？、得意なことはなんですか？
- ② 高校時代についての質問
 - ・ 高校時代に好きだった教科は？、部活はなんでしたか？、将来やりたいことは決まっていたか？
- ③ 仕事や勉強、データサイエンスについての質問
 - ・ 今の仕事や勉強に興味を持ったきっかけは？、今回の講座で特に相談に乗れることは？
- ④ 今後質問しやすくするための質問
 - ・ どのようなことを質問してよいですか？、どんな質問をされるとうれしいですか？
- ⑤ 少し楽しい質問
 - ・ たけのこ派ですか？きのこ派ですか？、自分を動物に例えるとなんですか？

④最強の紹介文を作しましょう

インタビューを踏まえ、最強の紹介文を作しましょう
今回のねらいも再確認しましょう

ねらい

- うちのメンターの最強の紹介文を作ること
- メンターにインタビューし、メンターの人柄や得意なこと、相談の仕方が伝わる自己紹介文を作しましょう
- 「最強」とは経歴がすごい！ではなく、困ったときに気軽に話しかけたいと思える自己紹介文が最強です

テンプレート

紹介文づくりに困ったら、例えば下記のような紹介文を参考にしましょう。自由に作成しても良いです。

「私たちのメンター、〇〇さんは、_____をしている人です。
特に、_____が得意で、今回の講座では_____について相談できます。
高校生の頃は、_____だったそうです。
実は、_____という一面もあります。
困ったときは、「_____」と声をかけると、_____してくれ
ます。
私たちにとって、_____なメンターです。」

作成した紹介文は暫定のものも含め、メモとしてTeamsにどんどん投稿してみましょう

SIGNATE Cloud/Google Colab のアクセス確認



SIGNATE Cloud

AIやデータサイエンスについて学べるe-Learning

https://user.cloud.signate.jp/sign_in

お気に入りに登録しましょう

学習画面例

The screenshot displays the SIGNATE Cloud interface with a dark theme. The top navigation bar includes links for 'Python入門', 'データ構造', 'データ操作 (tuple, list, dict)', and 'タプルの作成', along with a 'ブックマーク' button and user controls. The main content area is split into three panels: '説明' (Description), 'ソースコード' (Source Code), and '実行結果' (Execution Results).

説明 (Description): Explains how to create a tuple. It shows a table of items and their counts, and provides a code snippet for creating a tuple from a list.

商品名	金額	点数
ケーキ	300	3
パスタ	600	5
ステーキ	1000	2

```
In [2]: y = ("ケーキ", "パスタ", "ステーキ")
y
Out[2]: ("ケーキ", "パスタ", "ステーキ")
```

ソースコード (Source Code): Shows a Python script to create a tuple and print its elements.

```
1 # タプルを作ってみましょう
2 cake_tuple = (____,____)
3
4 # cake_tupleのデータ型を確認しましょう
5 print(____)
6
7 # タプルを出力してみましょう
8 print(____)
```

実行結果 (Execution Results): Currently empty.

At the bottom, there are buttons for '最初の状態に戻す' (Reset to initial state), '再試す' (Retry), and '学習する' (Learn).

忘れないように
メモしましょう

メールアドレス

パスワード

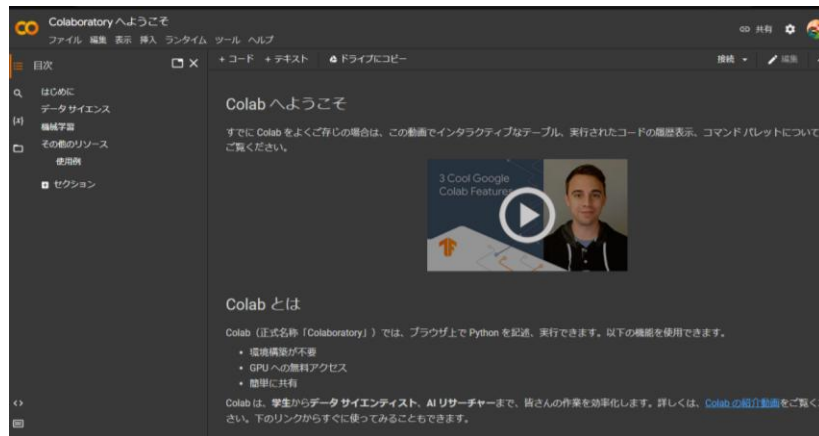
Google Colaboratoryとは？

Google Colaboratory



Googleが提供するnotebook

- ・ 環境設定不要でインターネットがあれば使える
- ・ GPUが無料で利用できる
- ・ 共有が簡単にできる



作業手順

<SIGNATE Cloudについて>

- ① 「SIGNATE Cloudからの招待」というメールを確認する
- ② メール中の「組織ユーザの登録」より、SIGNATE Cloudのアカウントを作成する
- ③ SIGNATE Cloudへログインできることを確認する
- ④ SIGNATE CloudのURLをお気に入りに登録する

<Google Colaboratoryの設定>

- ① Teamsに添付されているday1.zipをダウンロードする
- ② ダウンロードしたファイルを右クリックし、「全てを展開」よりzipファイルを解凍する
- ③ Google Driveにアクセスし、マイドライブ直下に「workspace」という名前でフォルダを作成する
※注意Google Driveは大文字小文字、スペースは別の名前扱いになりますので、「workspace」としてください
- ④ workspaceフォルダ内に、解凍した「day1」フォルダをアップロードする

これからやること

<Google Colaboratory>

- 回帰問題に挑戦してみよう
- 分類問題に挑戦してみよう

<Word>

- データ収集計画を立ててみよう

次回



次回について

本日設計したデータをアンケートで収集します。
ぜひ身の回りの方にも回答の協力をお願いしましょう！
集めたデータを使って自分だけのAIを作ってみよう！

7 日 月 火 水 木 金 土
2026 5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

2026 8月

9 日 月 火 水 木 金 土
2026 6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

日	月	火	水	木	金	土
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11 山の日	12	13	14	15
16	8月9日(日)@維新ホール					22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

◎ 次回までの宿題

・ AI入門 (2h)

⇒ <https://biz.quest.signate.jp/quests/10004>

・ 自動車環境性能の改善 (6h)

⇒ <https://biz.quest.signate.jp/quests/10001>

Appendix

操作方法など細かいマニュアル

Python



Pythonとは？



「読みやすく書きやすい」を目指したプログラム言語

- OS (windows, mac os等) が異なっても比較的容易に動かせる
- データ分析に必要なライブラリが豊富であり、R言語と同じようにデータ分析業務で好まれる
- 多くの機械学習フレームワークと互換性がある

※ライブラリとは便利なプログラムをまとめたもの

「読みやすく書きやすい」とはどういうこと？

- Pythonは、文法が非常にシンプルで、書き方が限られるため、**誰が書いても似たプログラムになりやすい** (=読みやすい)
- また他言語に比べて、**プログラムの記述量が非常に少なくすむ** (=書きやすい)

<サンプルプログラム> ※PythonとJavaの比較

- ①画面上に「Hello world!」と表示させるプログラム
- ②sample.txtというファイルから、データを1行ずつ読み込んで画面上に表示させるプログラム

```
print('Hello World!')
```

Python (①)

```
with open('sample.txt', 'r') as f:  
    for s in f.readlines():  
        print(s.strip())
```

Python (②)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Java (①)

```
import java.io.BufferedReader;  
import java.io.IOException;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
  
public class SampleJava {  
    public static void main(String[] args) {  
        Path path = Paths.get("sample.txt");  
        try(BufferedReader br = Files.newBufferedReader(  
            path)){  
            String str = "";  
            while((str = br.readLine()) != null) {  
                System.out.println(str);  
            }  
        } catch(IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

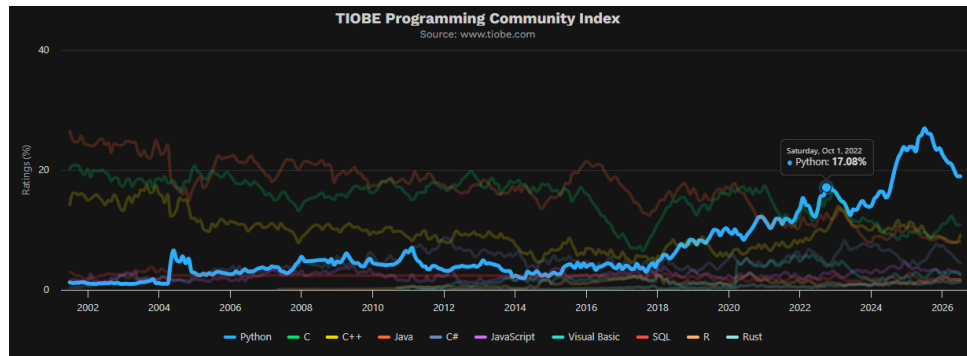
Java (②)

現在最も人気のプログラミング言語

- 2021年10月、プログラミング言語の人気・話題性を示すランキング（TIOBE PCI）でPythonが1位に
- そこから現在に至るまでずっと1位をキープ
- C言語、Java以外の言語が1位になるのは20年以上の歴史で初（！）

Jul 2026	Jul 2025	Change	Programming Language		Ratings	Change
1	1		 Python		18.94%	-8.03%
2	3	▲	 C		10.86%	+1.22%
3	2	▼	 C++		9.12%	-0.68%
4	4		 Java		8.03%	-0.73%
5	5		 C#		4.49%	-0.38%
6	6		 JavaScript		2.72%	-0.63%
7	8	▲	 Visual Basic		2.48%	+0.54%
8	13	▲▲	 SQL		1.71%	+0.32%
9	15	▲▲	 R		1.69%	+0.44%
10	18	▲▲	 Rust		1.34%	+0.33%

[出典] <https://www.tiobe.com/tiobe-index/>



Teamsの使い方

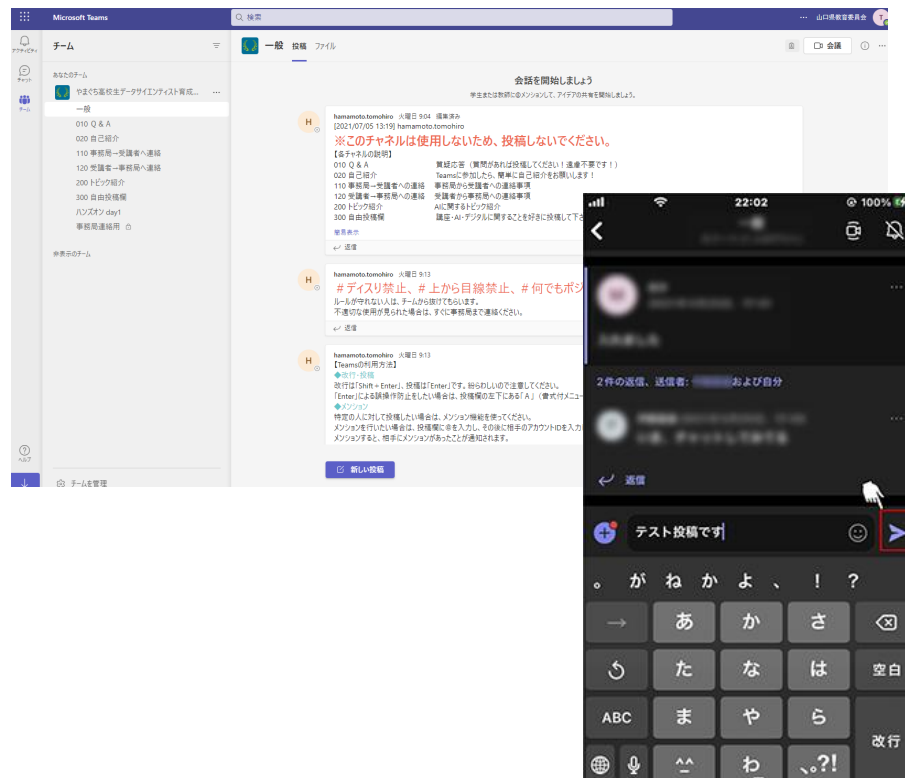


Teamsとは？

What is Teams ?

Microsoftが提供するコミュニケーションアプリです。

Teams を使うことで、リアルタイムなコミュニケーション、会議、共同作業、ファイル共有などが他メンバーと手軽にできます。

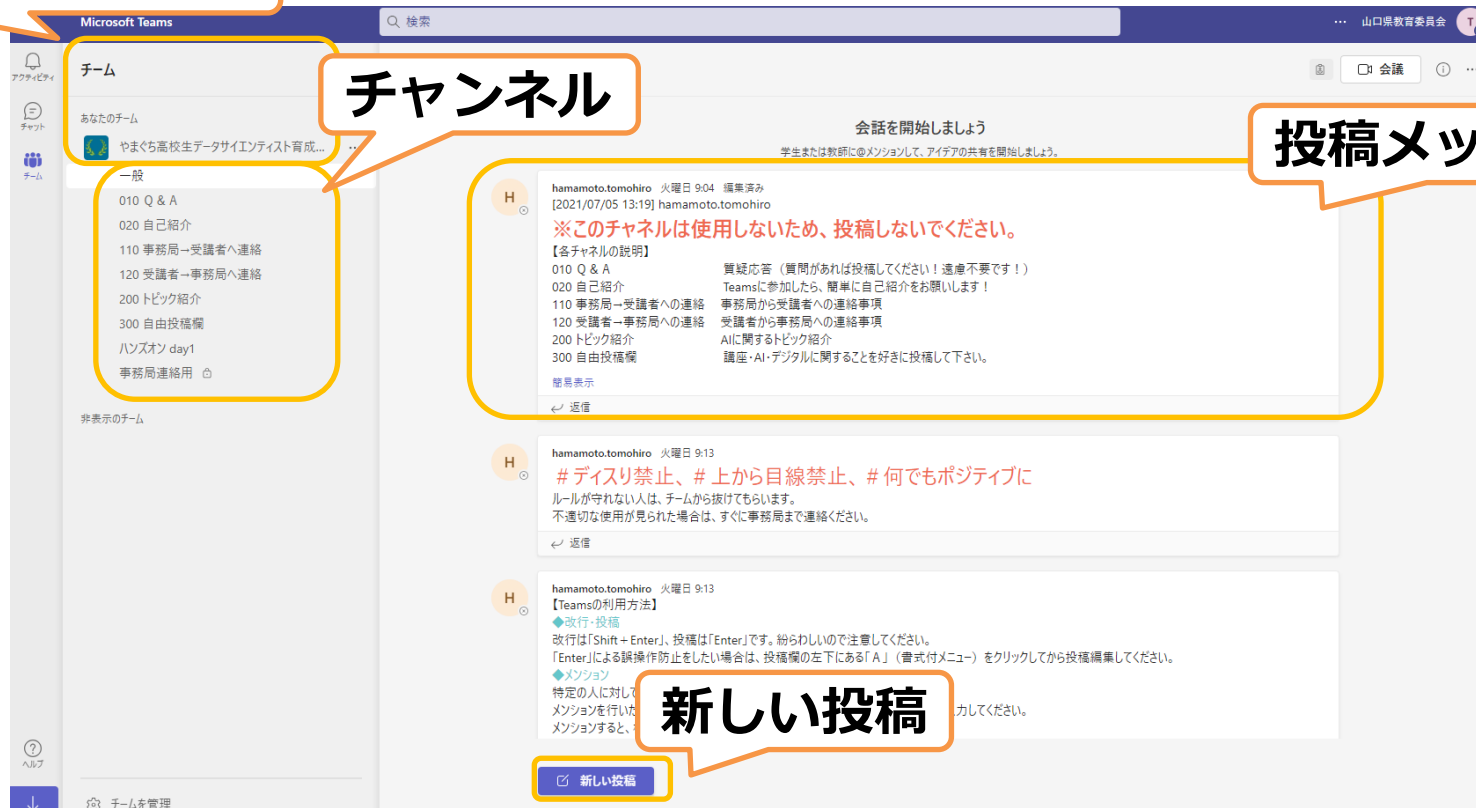


Teamsとは？

チーム

チャンネル

投稿メッセージ



Teamsとは？

会話を開始しましょう

学生または教師に@メンションして、アイデアの共有を開始しましょう。

hamamoto.tomohiro 火曜日 9:04 編集済み
[2021/07/05 13:19] hamamoto.tomohiro

※このチャンネルは使用しないため、投稿しないでください。

【各チャンネルの説明】
010 O & A 質疑応答（質問があれば投稿してください！遠慮不要です！）
[詳細表示](#)

返信

hamamoto.tomohiro 火曜日 9:13

ディスリ禁止、# 上から目線禁止、# 何でもポジティブに
ルールが守れない人は、チームから抜けてもらいます。
不適切な使用が見られた場合は、すぐに事務局まで連絡ください。

返信

hamamoto.tomohiro 火曜日 9:13

【Teamsの利用方法】
◆改行・投稿
改行は「Shift + Enter」、投稿は「Enter」です。紛らわしいので注意してください。
「Enter」による誤操作防止をしたい場合は、投稿欄の左下にある「A」（書式付メニュー）をクリックしてから投稿編集してください。

[詳細表示](#)

新しい投稿

1つの話題に返信をすることができます

※チャットは情報がすぐ流れてしまうため、1つの話題に対して議論を深めたい場合には返信を使って会話するのがベターです

Teamsとは？

スタンプ

コミュニケーションツールにおいて最も重要な機能だと思っています。LINEのように既読とはつかないため、このメッセージを見たのかどうかや、それに対する反応は**自発的**に行う必要があります。

いろいろな種類のスタンプがあるので、投稿があったら必ず反応しよう！



※slackの例

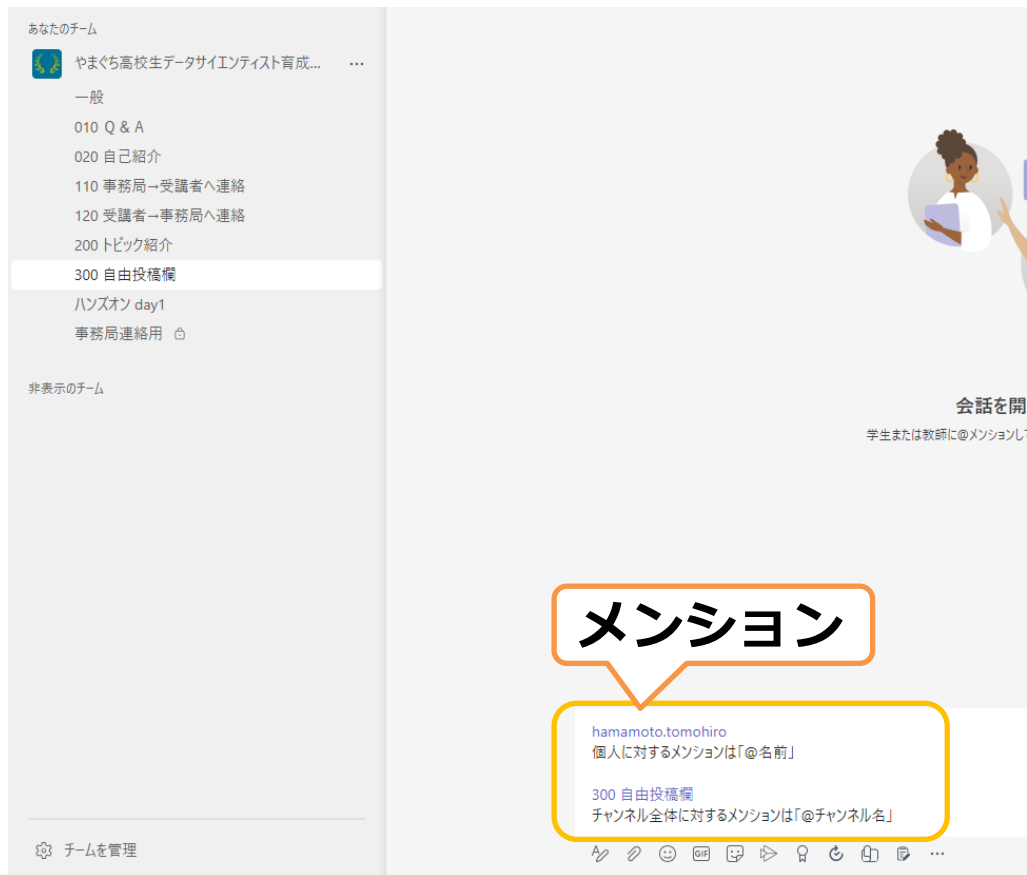


Teamsとは？

メンション

誰かに対するメッセージはメンションを付けることで、相手にメッセージがあったことをお知らせできます。

特定の誰かに向けたメッセージであれば必ずメンションを使いましょう。



@名前

→名前の人に通知

@channel

→チャンネルの人全員に通知

Teamsの書き方

マークアップ

デスクトップで Slack を使っている場合は、マークアップを使ってメッセージの書式を設定することもできます。

書式設定	説明
Bold	テキストをアスタリスクで囲みます： *対象のテキスト*
<i>Italicize</i>	テキストをアンダースコアで囲みます： _対象のテキスト_
Strikethrough	テキストの前後をチルダで囲みます： ~対象のテキスト~
<code>Code</code>	テキストをバッククォートで囲みます： `対象のテキスト`
Block quote	テキストの前に山括弧を追加します： >対象のテキスト
Code block	テキストの前に3つのバッククォートを追加します： ```対象のテキスト`
1. Ordered list	テキストの前に1とピリオドを追加して、 <code>スペース</code> キーを押します： 1. 対象のテキスト
Bulleted list	行をインデントするには、 <code>Tab</code> を押します。 テキストの前にアスタリスクを追加して、 <code>スペース</code> キーを押します： * 対象のテキスト 行をインデントするには、 <code>Tab</code> を押します。

よく使うもの

強調

`はshift+@

コードブロック

>引用

- 箇条書き①
- 箇条書き②

実際の画面

強調

```
a = 10
b = 20
c = 30
```

引用したいときに便利

- 箇条書き①
- 箇条書き②

-と箇の間にスペースが必要

Teamsとは？

文字のコミュニケーションは気を付けるべし。

文字のコミュニケーションでは皆さんご存じの通り、感情がわからないため、知らず知らずに相手のことを不愉快にしまうこともあります。

その為、投稿メッセージは

- ・ いろんな人が見ていることを意識する
- ・ なるべく丁寧な言葉遣いを選ぶ
- ・ スタンプをうまく使う

を心掛けたいですね。



Jupyter Notebookの使い方



Jupyter Notebook

NotebookのUIは次のようになっています。



Challenge MissionのUI

説明 ファイル一覧 データ内容

優良顧客を探せ！～顧客ターゲティング～

Challenge Missionでは、Questで取り組んだ定期預金キャンペーンの申込率予測にゼロからチャレンジします。
データはQuestで使用したものと同一データです。
また、機械学習モデルはQuestと同様、決定木を使用し、
random_stateは0としましょう。

データは、学習用データの `train.csv`、評価用データの `test.csv` および、予測結果を提出する際に必要となる `sample_submission.csv` が用意されています。
データの中身の詳細は、データ内容タブに記載しています。

なおデータはディレクトリ `../input` 以下にあります。例えば学習用データ `train.csv` を読み込む場合、以下のように記述します。

```
pd.read_csv("../input/train.csv")
```

予測ができれば、採点するための投稿用ファイルを作成します。
サンプルファイルが用意されていますので、まずサンプルファイルを読み込みます。

```
submit = pd.read_csv("../input/sample_submission.csv",  
header=None)
```

サンプルファイルはヘッダーがないため、引数 `header` に `None` を指定しています。
ファイルの内容としては、0列目にtestID、1列目にダミーデータが...

Save + Add ⌘ Cut ↑ Up ↓ Down ▶ Run ⏮ Run All ■ Stop Code ⌂ Restart Kernel

(unsaved changes)

In []:

notebook 投稿履歴

投稿する

Jupyter Notebook

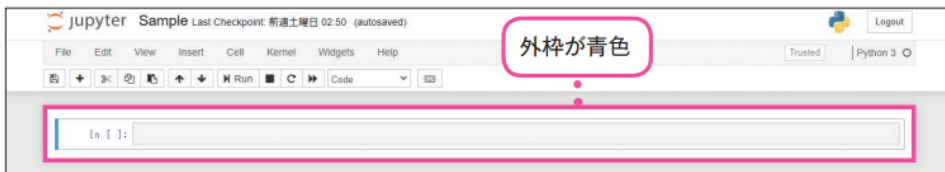
Notebookを利用するうえで重要になるのは「セル」です。

In[]の右にある入力欄を「セル」と呼び、このセルには「コマンドモード」と「編集モード」というモードが2種類あります。

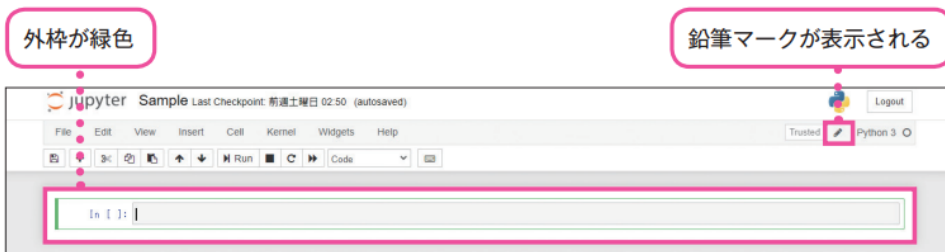
「コマンドモード」・・・セルを増やしたり減らしたり、操作するときのモード。外枠が青色。

「編集モード」・・・プログラムを入力するときのモード。外枠が緑色＋鉛筆マークが表示。

●コマンドモード



●編集モード



Jupyter Notebook

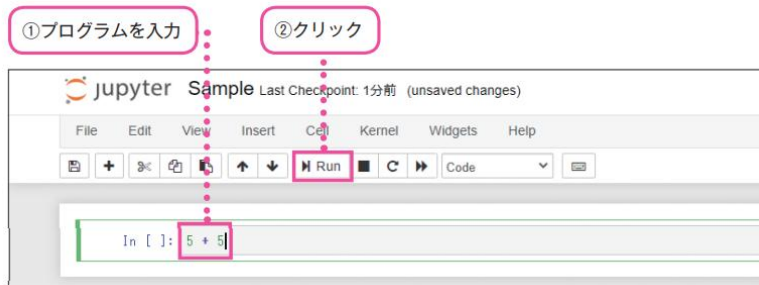
実際に動作の確認をしてみます。

セルをクリックすると、カーソルが表示され「編集モード」に切り替わります。

セル内に「5+5」を入力し、プログラムを実行してみてください。

プログラムの実行はツールバーの「Run」アイコンか、Shiftキーを押しながらEnterキーを押します。

後者の実行のショートカットキーは頻繁に利用するので、覚えておくと良いでしょう。



プログラムを実行すると、Out[1]の右側に実行結果が表示される。In[1]やOut[1]の数字はプログラムが何回目に実行されたかの履歴番号を表す。



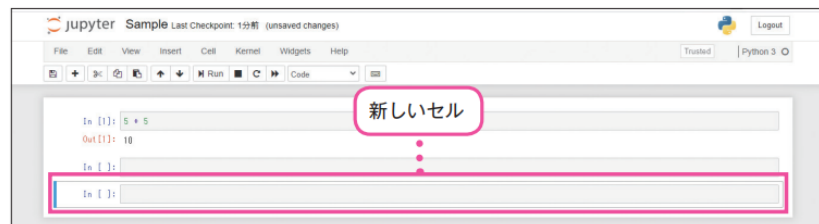
実行後、自動的に新規セルが作成されます。
この状態で入力欄以外をクリックするか、Esc
キーを押すと、コマンドモードに切り替わります



Jupyter Notebook

コマンドモードの状態ではセルを増やしたり減らしたりすることができます。
「B」キーを1回押すとセルが増え、「D」キーを2回押すとセルが削除されます。
「H」キーを押すと、ショートカットキーの一覧が表示されます。

●図 3-14 B キーによる新規セルの作成

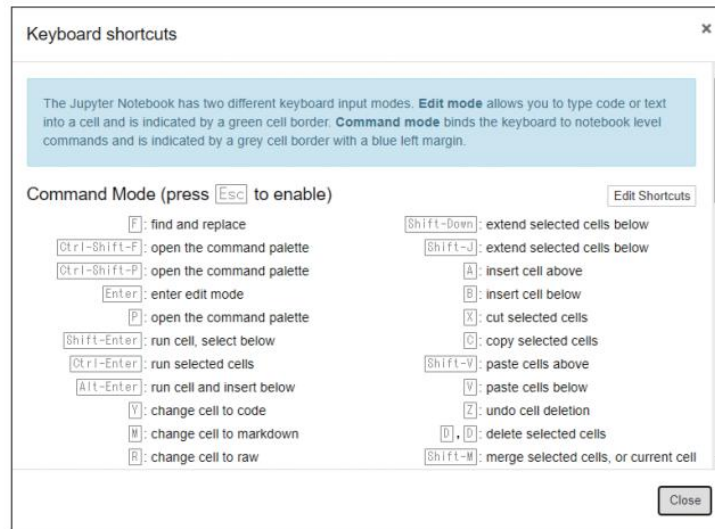


今度は「D」キーを2回押すと、セルが1つ削除されます。

●図 3-15 D キーの2回押しによるセルの削除



●図 3-16 H キーで表示されるショートカットキーの一覧



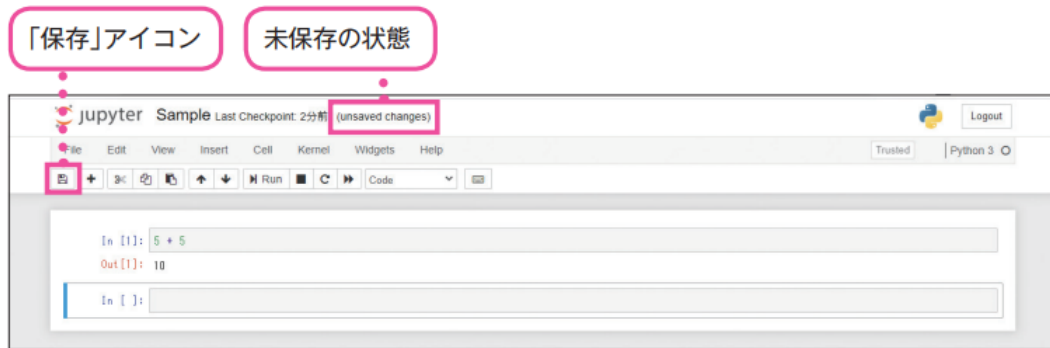
Jupyter Notebook

Notebookファイルの保存は、「保存」アイコンをクリックします。

Notebookには自動保存機能もあるため、画面の上部に「autosaved」が表示されていれば手動保存は必要ありません。未保存の場合には、「unsaved changes」と表示されますので、この時は手動保存しましょう。

なお、ブラウザとターミナルを終了すればJupyter Notebookは終了します。

Notebookファイルはプログラムのソースコードと実行結果については保存されますが、**一度終了したnotebookを再開したいときには、最初からプログラムを実行しなおす必要があります**。つまり、Notebookを終了すると、プログラムのソースコードや実行結果は残っていますが、ライブラリの宣言や変数の値はすべてリセットされてますので、再度の実行が必要です。その点についてよく注意しましょう。



Google Colaboratory

覚えてほしいショートカットキー

ショートカット内容	利用できるモード	ショートカットキー
セルを実行する	編集モード	Shift + Enter
編集モードからコマンドモード に切り替える	編集モード	ESC
コマンドモードから編集モード に切り替える	コマンドモード	Enter
セルを増やす	コマンドモード	Ctrl+M ⇒ B
セルを削除する	コマンドモード	Ctrl+M ⇒ D
削除したセルを元に戻す	コマンドモード	Z
ショートカットキー一覧を表示 する	コマンドモード	Ctrl+H

おしまい

Let's Enjoy DataScience!!!

