令和2年度「専修学校による地域産業中核的人材養成事業」

人工知能特論

令和2年度「専修学校による地域産業中核的人材養成事業」

人工知能特論

Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

目次

AIシステム科　　シラバス　　20XX年度入学生

科目名	人工知能特論					週合計駒数	1駒	作成日	
区分	必修 講義	開講時期	2年次 後期	週講義駒数 週実習等駒数	1駒 0駒	総時間数 総単位数	30時間 2単位	担当教員	

目標	概要
人工知能に関してどのような仕事が存在するのか学び、人工知能の各分野で議論されているトピックについて概要を理解する。	人工知能を活用したシステム構築に関わる人々のスキルセット、キャリアパスの事例について学習することで、社会における自身の活躍のイメージを詳細化する。

履修前提	テキスト・参考文献
※選択・エクステンションのみ記入	オリジナルテキスト

評価方法	関連科目
小テスト／中間テスト／期末テスト、提出課題、授業に取り組む姿勢（出席率、授業態度）	AIプログラミングⅠ・Ⅱ・Ⅲ、機械学習Ⅰ・Ⅱ・Ⅲ、人工知能概論、AIシステム開発

1	学習目標 人工知能構築プロジェクトに関わる職種について説明出来る。	学習項目 戦略コンサルタント、データサイエンティスト、データエンジニア、システムエンジニアなどの職種とスキルセットについて学習する。
	理解度確認：練習問題、小テスト	
2	学習目標 人工知能に関わる職種とスキルセット、キャリア構成について説明出来る。	学習項目 1の職種についてスキルセット、単価のレンジ、キャリア構成、日本と海外の事例について学習する。
	理解度確認：練習問題、小テスト	
3	学習目標 人工知能構築プロジェクトのステップについて説明出来る。	学習項目 人工知能構築プロジェクトの各ステップにおいてどのようなことをするのか学習する。
	理解度確認：練習問題、小テスト	
4	学習目標 これまでに学習した内容を復習し、理解を確実なものにする。	学習項目 これまでに学習した内容の理解を確実にするため、総復習を行う。
	理解度確認：演習問題	
5	学習目標 最新事例（データ収集）について説明出来る。	学習項目 データ収集の手法であるスクレイピングについて学習する。
	理解度確認：練習問題、小テスト	
6	学習目標 最新事例（音声認識）について説明出来る。	学習項目 音声認識技術の事例について学習する。
	理解度確認：練習問題、小テスト	
7	学習目標 最新事例（画像認識）について説明出来る。	学習項目 画像認識技術の事例について学習する。
	理解度確認：練習問題、小テスト	
8	学習目標 最新事例（回帰問題）について説明出来る。	学習項目 回帰問題アルゴリズムについて学習する。
	理解度確認：練習問題、小テスト	
9	学習目標 最新事例（時系列分析）について説明出来る。	学習項目 時系列分析のアルゴリズムについて学習する。
	理解度確認：練習問題、小テスト	
10	学習目標 最新事例（アンサンブル学習）について説明出来る。	学習項目 アンサンブル学習の考え方について学習する。
	理解度確認：練習問題、小テスト	
11	学習目標 最新事例（ニーズ予測）について説明出来る。	学習項目 マーケティングなどで用いられるニーズ予測について学習する。
	理解度確認：演習問題	

12	学習目標 最新事例(異常検知)について説明出来る。	学習項目 異常検知の事例について学習する。
	理解度確認: 練習問題、小テスト	
13	学習目標 最新事例(数理最適化)について説明出来る。	学習項目 数理最適化の事例と機械学習との組み合わせ事例について学習する。
	理解度確認: 練習問題、小テスト	
14	学習目標 最新事例(自然言語処理)について説明出来る。	学習項目 自然言語処理の事例について学習する。
	理解度確認: 演習問題	
15	学習目標 最新事例(グラフ理論)について説明出来る。	学習項目 グラフ理論の事例について学習する。
	理解度確認: 演習問題	

第1回：AIプロジェクトに関わる職種

職種とスキルセット

アジェンダ

- AIプロジェクト概要
- AIプロジェクトに関わる職種
  - 戦略コンサルタント
  - データサイエンティスト
  - データエンジニア
  - システムエンジニア

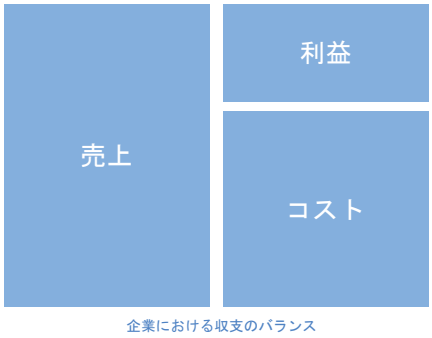
全15回の講義について

- 人工知能システム構築プロジェクトに関わる人々のスキルセット、キャリアパスの事例について学習することで、社会における自身の活躍のイメージを詳細化することを目的とします。

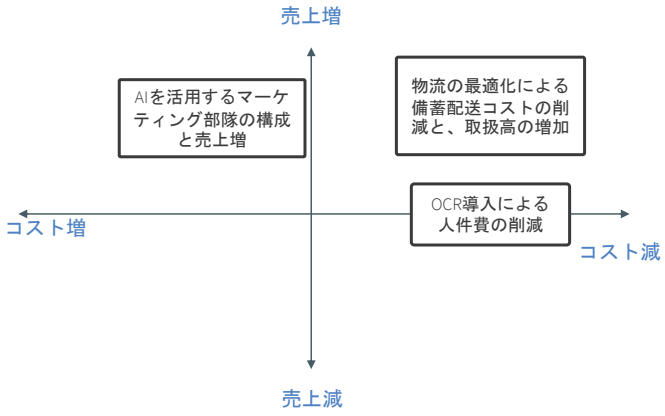
AIプロジェクト概要

AIを活用する視点：売上とコスト

企業活動の目的は利益を得ることですから、[売上>コスト]となる活動の手段としてAIを活用します。売上が増えてコストが減ることが最も望ましいのですが、売上が減ったとしてもコストをそれ以上に減らせるのであればAIを活用する意味はあります。

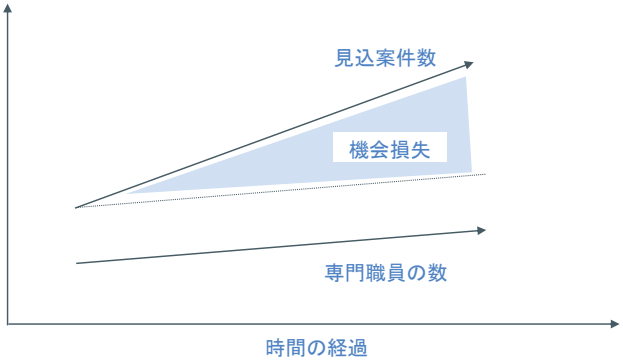


企業における収支のバランス



AIを活用する視点：時間軸

将来発生するであろう案件数が予測できると、将来的に不足するスキルと人材が予測できるようになります。人材の育成や調達には時間を要するため、将来の機会損失を減らすためには早めに見込み案件数を把握できたほうが好ましいです。



AIを活用する視点：競合他社との相対評価

自社の経営環境が参入障壁の高いものであればAIの導入は時間を掛けて検討できますが、参入障壁が低い業界であれば、競合他社と自社の相対評価がついて回ります。

他社が最新の手法を導入し業績が好調になったとします。自社の導入が遅れればシェアを奪われ、やがて衰退していくでしょう。

AIを活用する視点：人の能力の拡張

人には得意な行動/不得意な行動があるように、コンピュータアルゴリズムにも得意/不得意があります。
それぞれの得意/不得意を勘案し、役割分担をすることが重要です。

例えば金融市場における金融商品の取引を考えてみます。
HFT（high-frequency trading：高頻度取引）と呼ばれる取引は、非常に短い時間の間に何度も金融商品の取引が行われることです。金融商品の金額が表示される画面に張り付き、僅かな金額の上昇/下降を捉え続けるということは、人間には困難です。仮に画面を見続けられたとしても、瞬時に判断して取引を実行するのは物理的に不可能でしょう。このような作業は、コンピュータアルゴリズムに任せたほうがうまくいきます。

一方、ファンダメンタル投資と呼ばれる手法があります。これは企業の財務状態や経営環境の変化を考察し、HFTとは異なる長期的な視点で行う投資です。長期的に上げ下げがあるということは、現象を捉えるデータセットの数が少ないですし、そもそも景気の上げ下げは非常に複雑な現象で、この予測をコンピュータアルゴリズムで実行するというの困難です。このような分野は人間の経験や考察が大きな力を発揮する分野です。

AIプロジェクトのステップ

前ページまでで記載したように、AIの導入には案件それぞれの目的があります。AIを導入することでどのようなメリットが得られるのか、導入前によく検討する必要があります。

AIプロジェクトのステップには、このような効果を検討するステップを始め、AIの構築や運用まで多岐に渡るステップがあります。また、これらのステップを実行する職種があります。

職種ごとにどのようなことを担当するのか、以降のページで学習していきます。

AIプロジェクトに関わる職種

戦略コンサルタントの役割

コンサルティングファームにおいてパートナーやディレクターと呼ばれる職位にある人は、営業の責任を負うことが多いです。顧客との信頼関係を築き、プロジェクトの受注に結びつけ、コンサルティングファームの売上に貢献するという共同経営者という立場です。

顧客の課題は何か、どのような解決策があるのか、解決策を実行するにはどの様にすればよいのかということを、顧客と一緒になって考え実行します。
このステップの着地点が適切でなければ、それ以降のステップで投下するコストが無駄になってしまうため、非常に重要なステップです。
このように、AIプロジェクトにおける最上流のステップを実行する職種です。

データサイエンティストの役割

AIプロジェクトにおいては、技術的に高度な専門知識が要求されます。

「AI技術を用いて経営課題を解決する」青写真を描いたとしても、技術的に実現可能でなければ意味がありません。
これは、課題から出発して、その課題を解決する技術を組み合わせるというパターンです。
反対に、技術を広く理解していれば、目の前の課題について複数の解決策を策定することができます。これは技術から出発して解決できる課題を見つけるパターンです。

データサイエンティストは分析ができることが先ずは求められます。データサイエンティストが顧客のビジネスを理解し(ドメイン知識を獲得し)、実際にシステム化する際の運用やシステム設計にまで明るくなれば、転職市場での価値はどんどん高まるでしょう。

データエンジニアの役割

AIを活用したシステムだけによらず、何かしらのシステムはデータを加工して指標を作成し、指標を基に人間の判断を促す、という動きをします。
AIを活用したシステムでは、使用するモデルの特性に合わせてデータを維持管理する必要があります。
オープンデータを活用するのであれば、強固なデータ収集の仕組みも必要でしょう。

データの設計、データを取り扱うシステムの設計は、システムの拡張性・汎用性を左右する非常に重要な業務です。
このように、AIを活用するシステムが滞りなく動くために必要なデータの加工、運用などを広く扱う職種です。

システムエンジニアの役割

AIモデルとユーザーを橋渡しするフロントエンド、AIモデルとデータストレージの動きを制御するバックエンドと活躍の舞台が大別して2つあります。

フロントエンドのシステムエンジニアは、顧客の業務への深い理解が求められます。ドメイン知識とコミュニケーション力を武器として、市場価値を高めていけます。

バックエンドのシステムエンジニアはAIモデルの挙動への深い理解が求められます。また大規模データの取り回しなど、ドメインによらない汎用的なスキルを身につけることができ、技術力を武器として市場価値を高めていくことができます。

演習

演習1：AIを活用する視点（売上とコスト）

- AIシステムの事例を調査してください。
- 売上とコストの視点から導入の目的を説明してください。

演習2：AIを活用する視点（時間軸）

- AIシステムの事例を調査してください。
- 時間軸の視点から導入の目的を説明してください。

演習3：AIを活用する視点（競合他社との相対評価）

- AIシステムの事例を調査してください。
- 競合他社との相対評価の視点から導入の目的を説明してください。

演習4：AIを活用する視点（人の能力の拡張）

- AIシステムの事例を調査してください。
- 人の能力の拡張の視点から導入の目的を説明してください。

演習5：AIプロジェクトに関わる職種

- 皆さんが希望する職種は何でしょうか？本教材で取り扱った職種以外にも、様々な職種が存在します。
- 演習1～4で取り扱った事例から1つ取り上げ、皆さんが希望する職種はどのような成果を期待されているでしょうか？想像し、また皆さんの間で議論してみてください。

第2回：AIプロジェクトに関わる職種

スキルセットとキャリア構成

アジェンダ

- 前回の振り返り
- AIプロジェクトにおけるチームビルディング
- 職種ごとの単価の相場

前回の振り返り：AIプロジェクトに関わる職種

戦略コンサルタントの役割

コンサルティングファームにおいてパートナーやディレクターと呼ばれる職位にある人は、営業の責任を負うことが多いです。顧客との信頼関係を築き、プロジェクトの受注に結びつけ、コンサルティングファームの売上に貢献するという共同経営者という立場です。

顧客の課題は何か、どのような解決策があるのか、解決策を実行するにはどの様にすればよいのかということを、顧客と一緒になって考え実行します。
このステップの着地点が適切でなければ、それ以降のステップで投下するコストが無駄になってしまうため、非常に重要なステップです。
このように、AIプロジェクトにおける最上流のステップを実行する職種です。

データサイエンティストの役割

AIプロジェクトにおいては、技術的に高度な専門知識が要求されます。

「AI技術を用いて経営課題を解決する」青写真を描いたとしても、技術的に実現可能でなければ意味がありません。
これは、課題から出発して、その課題を解決する技術を組み合わせるというパターンです。
反対に、技術を広く理解していれば、目の前の課題について複数の解決策を策定することができます。これは技術から出発して解決できる課題を見つけるパターンです。

データサイエンティストは分析ができることが先ずは求められます。データサイエンティストが顧客のビジネスを理解し(ドメイン知識を獲得し)、実際にシステム化する際の運用やシステム設計にまで明るくなれば、転職市場での価値はどんどん高まるでしょう。

データエンジニアの役割

AIを活用したシステムだけによらず、何かしらのシステムはデータを加工して指標を作成し、指標を基に人間の判断を促す、という動きをします。
AIを活用したシステムでは、使用するモデルの特性に合わせてデータを維持管理する必要があります。
オープンデータを活用するのであれば、強固なデータ収集の仕組みも必要でしょう。

データの設計、データを取り扱うシステムの設計は、システムの拡張性・汎用性を左右する非常に重要な業務です。
このように、AIを活用するシステムが滞りなく動くために必要なデータの加工、運用などを広く扱う職種です。

システムエンジニアの役割

AIモデルとユーザーを橋渡しするフロントエンド、AIモデルとデータストレージの動きを制御するバックエンドと活躍の舞台が大別して2つあります。

フロントエンドのシステムエンジニアは、顧客の業務への深い理解が求められます。ドメイン知識とコミュニケーション力を武器として、市場価値を高めていけます。

バックエンドのシステムエンジニアはAIモデルの挙動への深い理解が求められます。また大規模データの取り回しなど、ドメインによらない汎用的なスキルを身につけることができ、技術力を武器として市場価値を高めていくことができます。

AIプロジェクトにおけるチームビルディング

AIプロジェクトにおけるチームビルディング

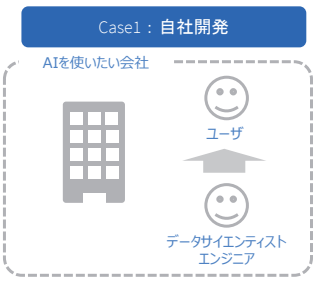
AI導入やデータ分析プロジェクトを遂行する場合、チームを組織することが一般的です。
プロジェクト遂行には様々なスキルが必要となりますし、作業ボリュームが1人でできる範囲ではないことがほとんどだからです。

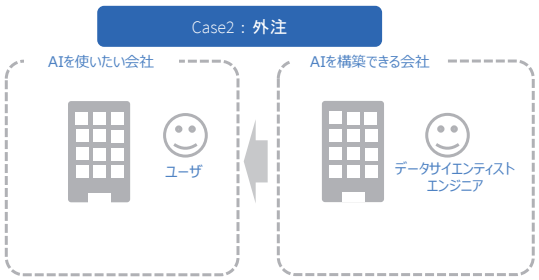
AI導入やデータ分析プロジェクトには、「できるかできないかやってみなければわからない」という類のプロジェクトが多数あります。
そのようなプロジェクトは、従来のシステム構築プロジェクトで採用されていたウォーターフォール型で実施されることはほとんどなく、「要件定義->分析->評価」を高速で繰り返すアジャイル型で実施されます。
チームメンバーがどのようなタスクを担っているのか、結果がどうだったのかをお互いに理解しあい、次の施策を決め、再び実行に移していきます。

そのような現場では、高いコミュニケーション能力とお互いの領域に対する理解・関心・敬意を持ちながらチームメンバーとして貢献することが求められます。

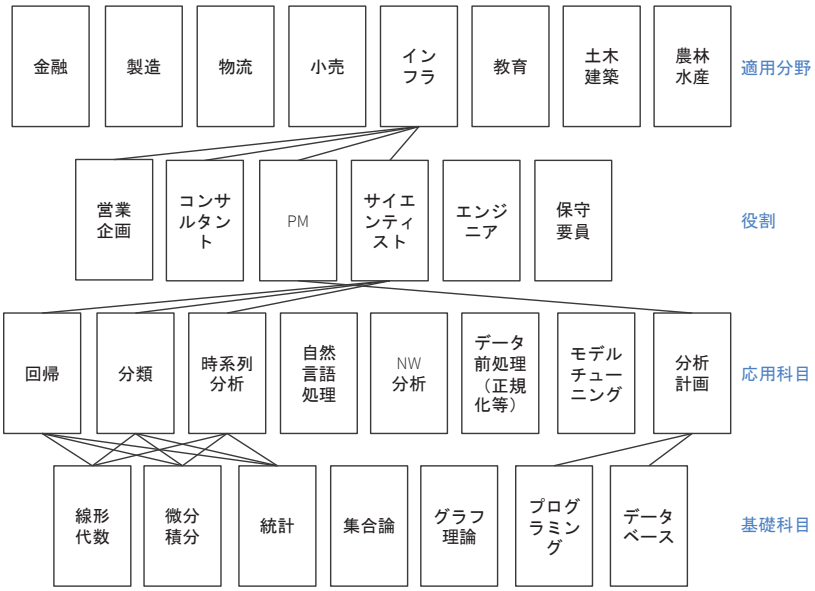
誰が実行するのか

- AIシステムの構築や分析プロジェクトを実施する場合、社内にAIに精通した人材がいれば自社開発・分析（Case1）を行います。社内にAIに精通した人材がいない場合、社外に発注（Case2）することになります。
- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に営業活動をすることになります。「AIを構築できる会社」の評判が良い場合、「AIを使いたい会社」からシステム構築を依頼されるケースもあります。





AIプロジェクトにおける役割・スキルセットの例

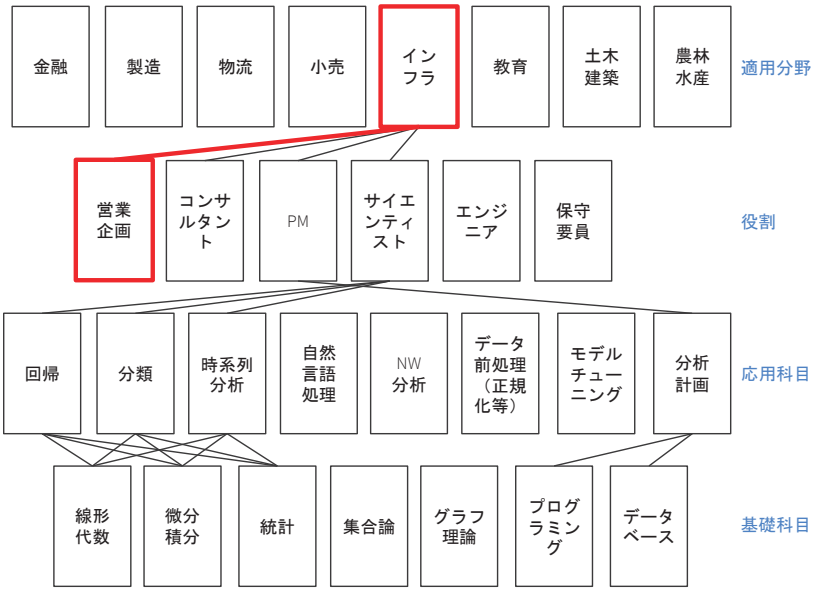


ある架空のプロジェクトを例として、プロジェクトメンバーの選定が行われる過程を説明していきます。

顧客はインフラを構築整備する企業であるとします（全ページのCase2に該当し、みなさんはAIを構築する会社の社員だとします）。
公共性の高いインフラですので、国や地方自治体の補助金で構築費用が賄われることがあります。一方、運用は地方自治体の独立採算が求められることがあります。

このようなビジネス環境で企業活動をしている顧客が、「自社の保有するデータを活用して全社的な利益率の向上を図りたい」と依頼してきたとします。

AIプロジェクトにおける役割・スキルセットの例

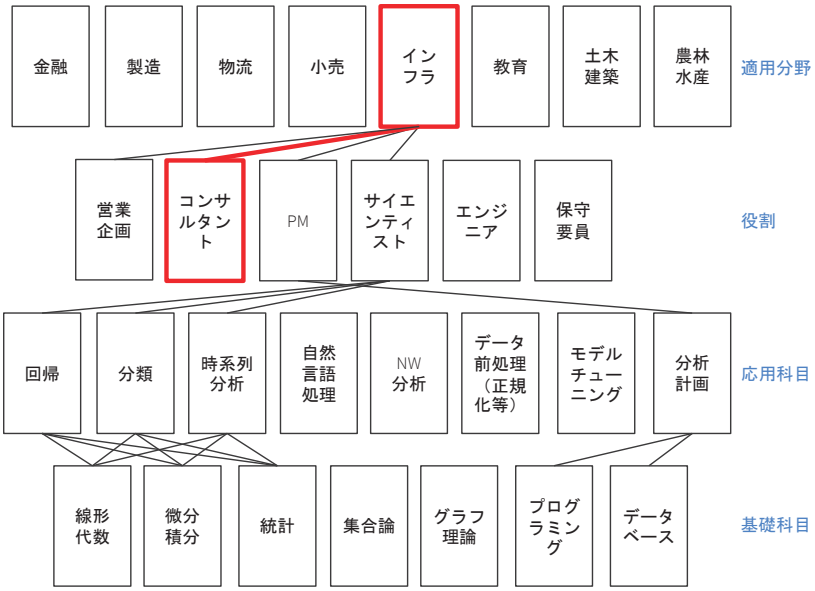


まず、営業企画をするメンバーの存在があります。
AI構築や分析を生業とする同業他社が増えている状況で、どのような分野に自社が進出すれば、参入障壁を高めて安定したポジションを築くことができるのか、常に自社の戦略を練り、業界にアンテナを張り巡らせています。

インフラは人間が経済活動をする限り存続するもので、もしインフラを生業とする顧客のビジネスパートナーとなれれば、自社にとっては安定に繋がります。

この戦略を実行に移した営業企画のメンバーが、この案件を受注しました。

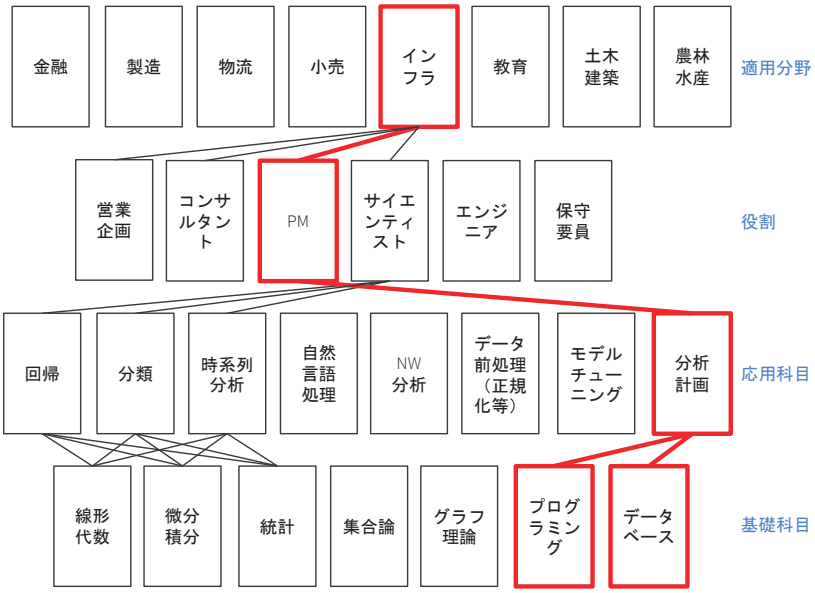
AIプロジェクトにおける役割・スキルセットの例



顧客が構築するのは公共性の高いインフラですので、国や地方自治体の補助金で構築費用が賄われることがあります。一方、運用は地方自治体の独立採算が求められることがあります。

このような特殊なビジネス環境で企業活動を行っている顧客のデータには、業界特有の事象が随所に現れてきます。
この特殊なデータが意味することを読み解き、顧客の利益率を向上させるためには、業界に精通しているコンサルタントのドメイン知識が必要となります。

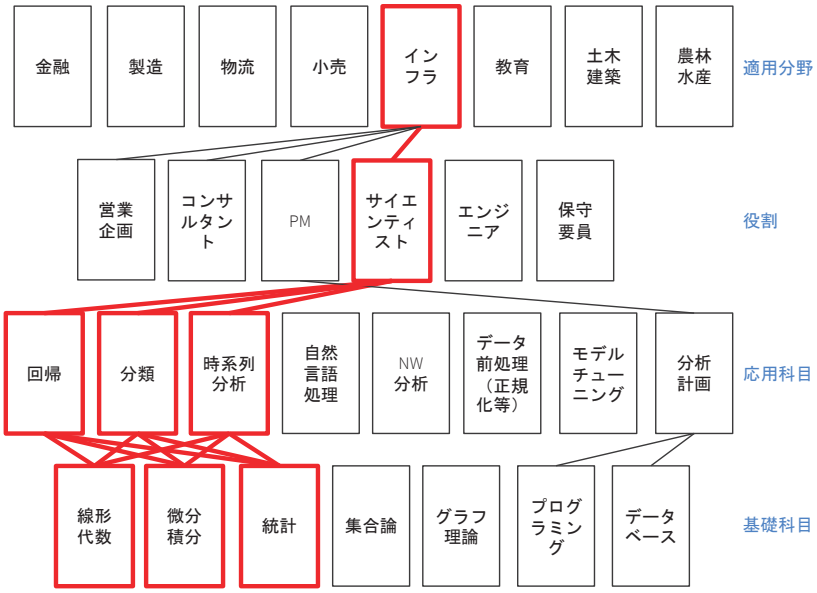
AIプロジェクトにおける役割・スキルセットの例



プロジェクト遂行するにはマネージャーが必要になります。
顧客の要望に答えるため、どのような分析をどの順番で実行すれば最も費用対効果が高いのか、分析の計画を立てます。

分析の計画は実現可能性があるものでなければならないので、プロジェクトマネージャーには技術に対する理解も求められます。

AIプロジェクトにおける役割・スキルセットの例



顧客のデータを実際に分析し、考察を加えて、利益率の向上を図るというシナリオを構築する必要があります。

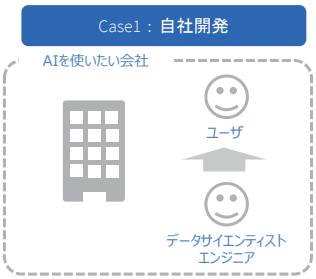
このようなプロジェクトは黒字/赤字になりやすい、このような遷移をするプロジェクトは黒字/赤字になりやすいなど、分類/回帰/時系列問題を組み合わせて分析する必要があります。

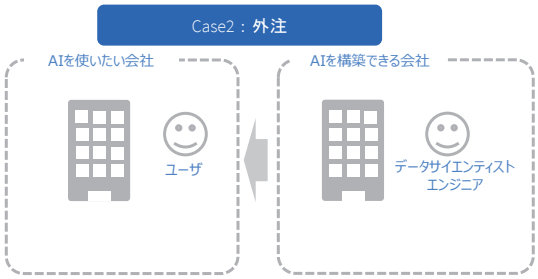
この様に、顧客の要望を実現するための分析技術を持つデータサイエンティストが必要となります。

職種ごとの単価の相場

AIシステム構築の対価

- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に対してAIシステム構築/分析サービスを実施し、対価をもらうことになります。
- プロジェクトに従事するメンバーの職種ごとに単価の相場があり、単価×稼働時間で対価を計算することが一般的です。
- 次ページ以降で、職種ごとの単価の目安値について説明していきます（※企業規模、外資/内資、企業のブランドによって金額は大きくばらつきますので、あくまで目安の値と捉えてください）。





職種ごとの単価の相場

パートナークラスのコンサルタントとなると、時間あたり5～10万円ほど請求することもあります。
マネージャークラスで時間あたり2～3万円程度、アソシエイトコンサルタントでも時間あたり1万円程度請求することがあります。

実際には、単価の高いパートナーやマネージャークラスが一つのプロジェクトにフルアサインすることはなく、月あたり500万円程度に抑えることが多いようです。
現場に張り付きで稼働するシニアコンサルタントやアソシエイトコンサルタントは、月当たり300～100万円程度となります。

職種ごとの単価の相場

チームをリードするデータサイエンティストは月当たり500～300万円ほど、メンバークラスのデータサイエンティストで月当たり200～100万円程度請求することになります。

データサイエンティストもコンサルタント同様に、単価の高い職位のメンバーは稼働を抑え、現場に張り付く月単価のメンバーを増やし、全体の金額を上げていくことが多いです。

※
企業のブランドによって、単価は大幅に上下します。
米国シリコンバレーに本社を構える某IT企業のデータサイエンティストとお話したことがあるのですが、その企業はデータサイエンティスト一人あたり月当たり1,000万円を顧客に請求するそうです。

職種ごとの単価の相場

データエンジニアやAI案件に携わるシステムエンジニアの相場は月当たり150～70万円程度となるようです。

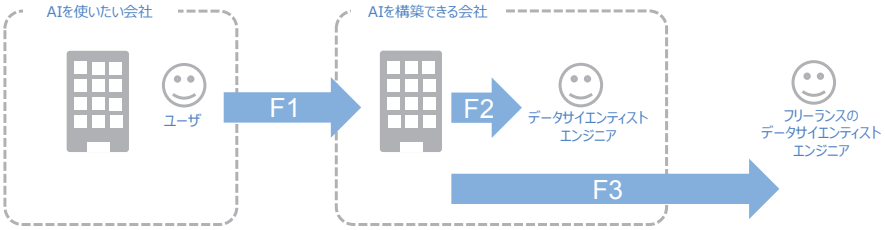
基幹系業務システムなどの大型案件でキャリアを築いてきたエンジニアがAIやデータ分析系キャリアに変更するときに、データエンジニアやシステムエンジニアは飛び込みやすい職種なので、自然とエンジニア人口が多くなります。供給が多くなり価格競争が起きやすい構図となっています。

商流について

AIやデータ分析に関するポジションは様々です。

- 「AIを使いたい会社」の社内にデータサイエンス部隊があり、そこで働く。
- 「AIを構築できる会社」で働く（前ページまでで扱ってきた単価はここに相当します）。
- フリーランスとして働く。

商流はシンプルな方が自分に入ってくる対価は多くなりますが、営業/事務作業コストも大きくなります。
対価が多くなるということは当然ながら成果物の品質をシビアに見られることになり、プレッシャーが大きくなります。
キャリアプランを立てるときは、自身の経験や人脈をしっかりと整理し、リスクを抑えてつつチャレンジしていくことが行くことが肝要でしょう。



演習

演習1：営業・企画の役割

- あなたはあるシステム構築会社の営業企画職に就いているとします。どのような分野のプロジェクトを受注したいでしょうか？
- その分野の企業に営業に行く前に、業界の事前調査をすることにしました。あなたはどのような切り口で調査するでしょうか？

演習2：コンサルタントの役割

- 営業企画のチームメンバーから、一緒に顧客を往訪してくれと要請がありました。
- プロジェクトが受注間近なので、受注後に速やかにプロジェクトを開始できるように、顧客のことを知ってほしいとのことでした。
- 一週間後に、コンサルタントであるあなたは顧客を訪ねることになります。あなたはどのような準備をして臨むでしょうか？

演習3：データサイエンティストの役割

- データサイエンティストであるあなたは、コンサルタントとともに要件定義をしています。
- コンサルタントのチームメンバーは業務知識には明るいのですが、分析技術や数理ロジックには明るくありません。あなたは、どのようにチームメンバーと協力するでしょうか？

演習4：データエンジニアの役割

- データエンジニアであるあなたは、要件定義からプロジェクトに参画しました。
- 顧客がやりたいことを実現するために、データサイエンティストであるあなたのチームメンバーは、市場のトレンドが反映されたデータの分析をおこなっていました。
- あなたはデータサイエンティストから相談を受けました。取り扱うデータの特性上、トレンドが変化するととたんにモデルは精度が低下するそうです。モデルをリリースした後に精度が著しく低下した場合、すぐにモデルを切り戻すシステム構成にできないか、という相談でした。
- あなたが設計するシステムは、どのような機能が必要でしょうか？

第3回：AIプロジェクトのステップ

アジェンダ

- 前回までの振り返り
- AIシステム構築プロジェクトのステップ

前回までの振り返り：AIプロジェクトに関わる職種

戦略コンサルタントの役割

コンサルティングファームにおいてパートナーやディレクターと呼ばれる職位にある人は、営業の責任を負うことが多いです。顧客との信頼関係を築き、プロジェクトの受注に結びつけ、コンサルティングファームの売上に貢献するという共同経営者という立場です。

顧客の課題は何か、どのような解決策があるのか、解決策を実行するにはどの様にすればよいのかということを、顧客と一緒になって考え実行します。
このステップの着地点が適切でなければ、それ以降のステップで投下するコストが無駄になってしまうため、非常に重要なステップです。
このように、AIプロジェクトにおける最上流のステップを実行する職種です。

データサイエンティストの役割

AIプロジェクトにおいては、技術的に高度な専門知識が要求されます。

「AI技術を用いて経営課題を解決する」青写真を描いたとしても、技術的に実現可能でなければ意味がありません。
これは、課題から出発して、その課題を解決する技術を組み合わせるというパターンです。
反対に、技術を広く理解していれば、目の前の課題について複数の解決策を策定することができます。これは技術から出発して解決できる課題を見つけるパターンです。

データサイエンティストは分析ができることが先ずは求められます。データサイエンティストが顧客のビジネスを理解し(ドメイン知識を獲得し)、実際にシステム化する際の運用やシステム設計にまで明るくなれば、転職市場での価値はどんどん高まるでしょう。

データエンジニアの役割

AIを活用したシステムだけによらず、何かしらのシステムはデータを加工して指標を作成し、指標を基に人間の判断を促す、という動きをします。
AIを活用したシステムでは、使用するモデルの特性に合わせてデータを維持管理する必要があります。
オープンデータを活用するのであれば、強固なデータ収集の仕組みも必要でしょう。

データの設計、データを取り扱うシステムの設計は、システムの拡張性・汎用性を左右する非常に重要な業務です。
このように、AIを活用するシステムが滞りなく動くために必要なデータの加工、運用などを広く扱う職種です。

システムエンジニアの役割

AIモデルとユーザーを橋渡しするフロントエンド、AIモデルとデータストレージの動きを制御するバックエンドと活躍の舞台が大別して2つあります。

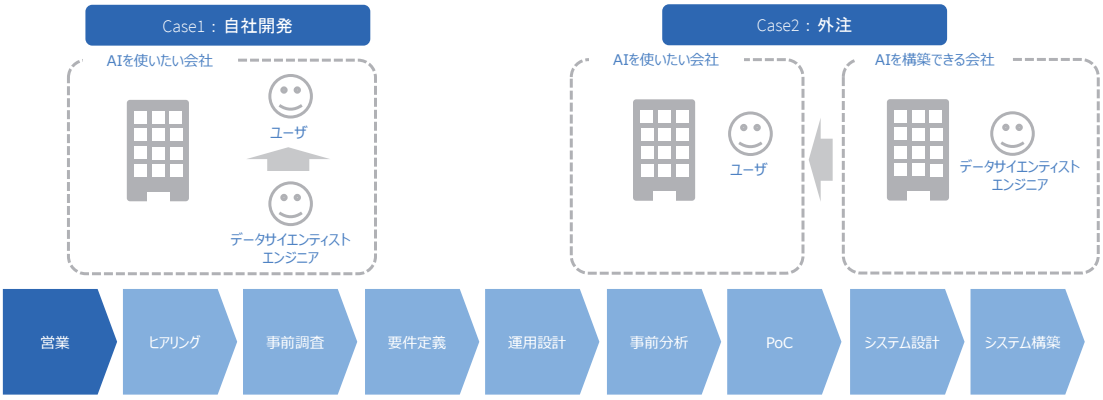
フロントエンドのシステムエンジニアは、顧客の業務への深い理解が求められます。ドメイン知識とコミュニケーション力を武器として、市場価値を高めていけます。

バックエンドのシステムエンジニアはAIモデルの挙動への深い理解が求められます。また大規模データの取り回しなど、ドメインによらない汎用的なスキルを身につけることができ、技術力を武器として市場価値を高めていくことができます。

AIシステム構築プロジェクトのステップ

営業

- AIシステムを構築する場合、社内にAIを構築できる人材がいれば自社開発（Case1）を行います。社内にAIを構築できる人材がいない場合、社外に発注（Case2）することになります。
- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に営業活動をすることになります。「AIを構築できる会社」の評判が良い場合、「AIを使いたい会社」からシステム構築を依頼されるケースもあります。



Case1：自社開発の事例

自社サービス開発を、自社のデータサイエンティスト/アナリストで実施しているケースです。
自社サービスで発生するデータを自由な発想で分析できることや、人月単価に縛られない価値を発揮できるなど、魅力的な環境だと思います。

参照：https://careers.mercari.com/jp/job-categories/analyst/

参照：https://linecorp.com/ja/career/position/588

Case2：他社のAIシステムを開発する事例

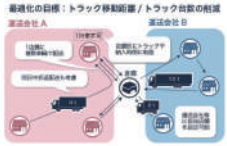
顧客のサービスやシステムを開発する事例です。
分野を横断してノウハウを習得することができます。いち早く有用な技術や方法論を習得し横展開することで、顧客から非常に重宝されます。

メガバンク初 AIによる住宅ローン事前審査　高度なスキルとAIの融合で、審査時間を約1日から最短15分へ短縮

株式会社三菱UFJ銀行様

金融

参照：https://jpn.nec.com/ai/case/index.html



業種　物流

活用技術　データ分析

属人化していた配車計画の立案業務を、数理最適化技術を用いて自動化し、ルートや配車台数などの最適化によりコスト削減を実現。

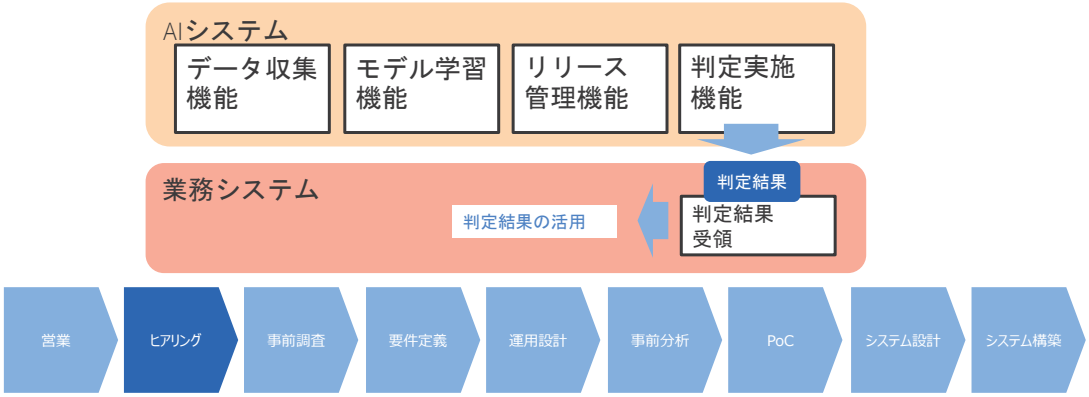
参照：https://ai.brainpad.co.jp/case-study/

AI導入で製品検査を効率化。生産ラインの変化・変動に即応できる画像認識システムとは？

参照：https://www.fujitsu.com/jp/solutions/business-technology/ai/ai-zinrai/customer-stories/

ヒアリング

- AIを使いたい会社・ユーザはAIを使って何を実現したいのか、ヒアリングを行います。
- 特に既存の業務システムとの連携を考えている場合は、業務システムの仕組み上実現が難しい場合や費用が大きくなる場合もあるため、導入コストと得られるメリットの双方を勘案する必要があります。
- AIを導入しなくてもユーザのやりたいことが実現できると判断する場合や、コストが大きすぎて導入を避けた方や良い場合は、プロジェクト自体の中断を決断することになります。



事前調査

- AI導入にメリットがあると判断した場合、データ・技術の事前調査へと進めます。
- 機械学習などを用いてAIモデルを作成するためには、データの量と質がモデルを構築するに足るかどうかを判断する必要があります。

- データの量と質が基準を満たしている場合でも個人情報が含まれていたり、コンプライアンス（法令遵守）の視点からデータの使用が困難であったりする場合があります。コストを投入する前に様々な角度からプロジェクトの実現性を検証する必要があります。

要件定義

- まずはシステムが満たすべき条件を決定します。例えばモデルのインプットデータの項目と量、モデルのアウトプットの項目、モデルの精度などです。
- モデル以外にも蓄積するデータの量と、蓄積の速度・期間などについて決定します。

- システム以外にも、サービスとして満たすべき条件を決定します。例えばユーザーからのリクエストに対して何秒以内に応答を返さないといけないか、一日何回の使用に耐えなければならないか、システム運用のコストと売上の損益分岐をどこに定めるかなどです。

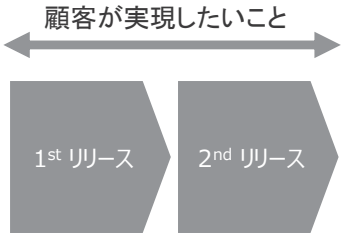
- 実務においては後述する運用設計・事前分析・PoCを数回繰り返して要件を決定していきます（実際にやってみて発覚することがあるため、このような繰り返しのやり方が必要となります）。

「スコープを切る」

顧客が実現したいことをシステムなどで具現化する際に、あえて段階的に実施することがあります。

仕事を受ける側からすると、一度に受注したほうが売上が大きくなるという誘惑に駆られますが、失敗したときのダメージが大きくなります。
一度に全てのことを実現するには複雑すぎてリスクが見通せない場合や、あえて踊り場を設けて実験を繰り返しながら着実に積み上げる場合など、スコープを切る理由は様々です。

変化に対応しようという意欲が高くなるほど「あれもこれも」と前のめりになってしまいます。多くの事例を経験し、きちんとリスクを顧客に伝えることができる人材は、どの現場でも重宝されます。



運用設計

- AIモデルを実サービスとして展開し続けるためには、運用が必要になります。
- 増え続けるデータの蓄積、モデルの更新、不具合への対応にどれだけの人・機材・お金・時間を投入するのかを決定します。投入するこれらのコストを勘案し、ユーザに請求するサービス料の設定が必要となります。

- ユーザの予算が決まっていて多額の投資をできない場合、システム機能の削減や運用フローの簡素化を実施します。

「運用設計」はいつ実施するのが適切か？

本資料では、「運用設計」は「事前分析」や「PoC」の前のステップとして記載されています。

「やってみなければできるかどうかわからない」というAI案件は、PoC（概念実証）という実現可能性を検証するステップを置くことがあります。

PoCでは素晴らしいモデルが構築され良い結果が出たとしても、いざ実システムの設計や運用を考える段階になって、「実現するのは難しい」という結論になってしまうことが少なくありません。
これはPoCのために特別に仕立てたデータを使用していたり、個人情報保護やセキュリティの問題をPoCの最中だけ特別にクリアできる条件で実施していたりと、本来の条件ではない条件で実施することがあるからです。

運用設計を早い段階で実施することで、「本当に実現できるのか？」ということを早めに熟慮せざるを得なくなります。
早い段階でだめだとわかれば、その後投下する予定だったヒト・モノ・カネを節約することができます。

事前分析

- モデル作成に使用するデータの詳細な調査を実施します。
- モデルの目的変数、説明変数の統計値を算出したり、目的変数と説明変数間で相関があるのかなどの検証を実施します。

- 本プロセスにおいてデータの大きな偏りが確認されたり、また説明変数と目的変数間に全く相関が確認されない場合、機械学習を導入しても効果が見られないと判断してプロジェクトを中断することもあります。このステップはプロジェクトの「勝ち目」を判断する重要なステップです。

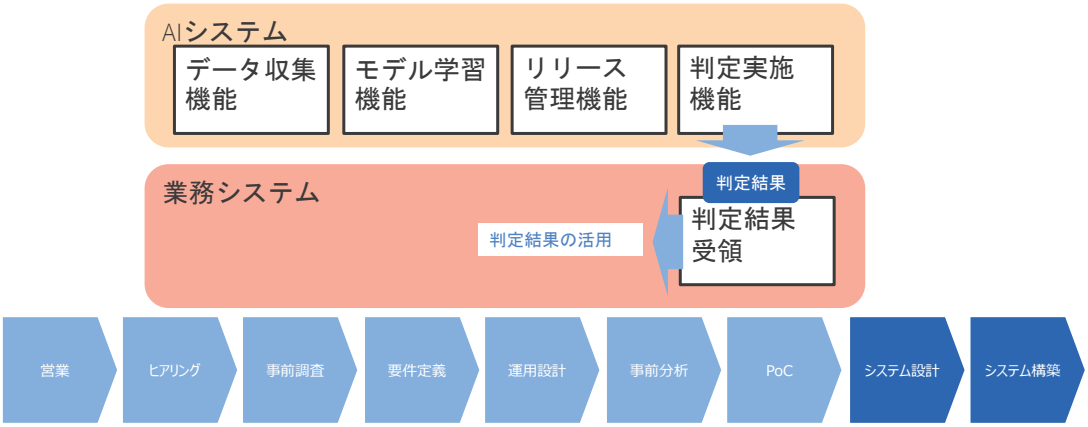
PoC（Proof of Concept：概念実証）

- AIモデルが本当に実現できるのか、実際に試してみるステップです。
- データ加工、機械学習アルゴリズムの試行、精度検証をいかに正確に積み上げていけるかがプロジェクト成否の分かれ目となります。機械学習の実施が効率化できるライブラリの使用や、独自スクリプトの構築が欠かせません。

- このステップで大事なのはモデルの精度向上ではありますが、要件定義で決定した事項を満たせているのか、ということを常にチェックしなければなりません。例えば「精度の向上を優先しすぎるあまりDeep Neural Networkをアルゴリズムとして選択してしまい、モデルの説明性が低下する。」という事態を避けなければなりません。

システム設計・システム構築

- モデルを作成した後は、データ収集からモデル判定結果の提供方法までを含めたシステム設計が必要となります。
- 設計後は実際にシステムを構築していきます・



演習

演習1：AIシステム構築の主体

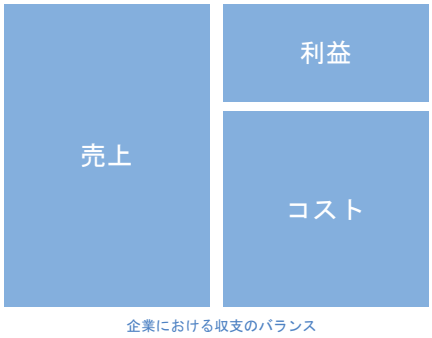
- 自社サービス/システムを自社のデータサイエンティスト部隊で開発している事例を調査してください。
- 顧客のサービス/システムを構築する事例を調査してください。
- あなたが参画するとしたら、どちらのケースが良いでしょうか。その理由について整理してください。

第4回：人工知能特論総復習

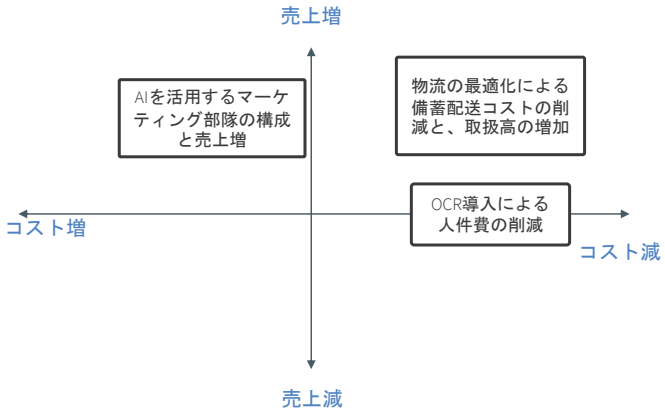
第1回：AIプロジェクトに関わる職種

AIを活用する視点：売上とコスト

企業活動の目的は利益を得ることですから、[売上>コスト]となる活動の手段としてAIを活用します。売上が増えてコストが減ることが最も望ましいのですが、売上が減ったとしてもコストをそれ以上に減らせるのであればAIを活用する意味はあります。

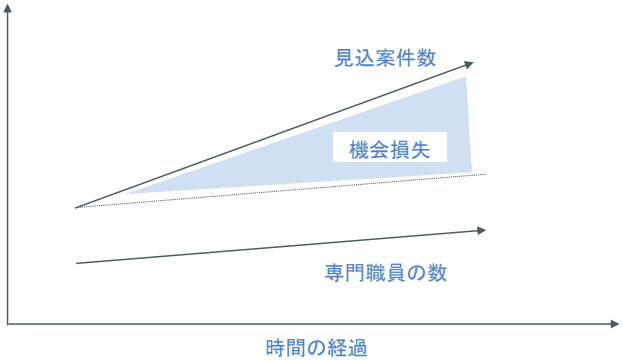


企業における収支のバランス



AIを活用する視点：時間軸

将来発生するであろう案件数が予測できると、将来的に不足するスキルと人材が予測できるようになります。人材の育成や調達には時間を要するため、将来の機会損失を減らすためには早めに見込み案件数を把握できたほうが好ましいです。



AIを活用する視点：競合他社との相対評価

自社の経営環境が参入障壁の高いものであればAIの導入は時間を掛けて検討できますが、参入障壁が低い業界であれば、競合他社と自社の相対評価がついて回ります。

他社が最新の手法を導入し業績が好調になったとします。自社の導入が遅れればシェアを奪われ、やがて衰退していくでしょう。

AIを活用する視点：人の能力の拡張

人には得意な行動/不得意な行動があるように、コンピュータアルゴリズムにも得意/不得意があります。
それぞれの得意/不得意を勘案し、役割分担をすることが重要です。

例えば金融市場における金融商品の取引を考えてみます。
HFT（high-frequency trading：高頻度取引）と呼ばれる取引は、非常に短い時間の間に何度も金融商品の取引が行われることです。金融商品の金額が表示される画面に張り付き、僅かな金額の上昇/下降を捉え続けるということは、人間には困難です。仮に画面を見続けられたとしても、瞬時に判断して取引を実行するのは物理的に不可能でしょう。このような作業は、コンピュータアルゴリズムに任せたほうがうまくいきます。

一方、ファンダメンタル投資と呼ばれる手法があります。これは企業の財務状態や経営環境の変化を考察し、HFTとは異なる長期的な視点で行う投資です。長期的に上げ下げがあるということは、現象を捉えるデータセットの数が少ないですし、そもそも景気の上げ下げは非常に複雑な現象で、この予測をコンピュータアルゴリズムで実行するというの困難です。このような分野は人間の経験や考察が大きな力を発揮する分野です。

AIプロジェクトのステップ

前ページまでで記載したように、AIの導入には案件それぞれの目的があります。AIを導入することでどのようなメリットが得られるのか、導入前によく検討する必要があります。

AIプロジェクトのステップには、このような効果を検討するステップを始め、AIの構築や運用まで多岐に渡るステップがあります。また、これらのステップを実行する職種があります。

職種ごとにどのようなことを担当するのか、以降のページで学習していきます。

戦略コンサルタントの役割

コンサルティングファームにおいてパートナーやディレクターと呼ばれる職位にある人は、営業の責任を負うことが多いです。顧客との信頼関係を築き、プロジェクトの受注に結びつけ、コンサルティングファームの売上に貢献するという共同経営者という立場です。

顧客の課題は何か、どのような解決策があるのか、解決策を実行するにはどの様にすればよいのかということを、顧客と一緒になって考え実行します。
このステップの着地点が適切でなければ、それ以降のステップで投下するコストが無駄になってしまうため、非常に重要なステップです。
このように、AIプロジェクトにおける最上流のステップを実行する職種です。

データサイエンティストの役割

AIプロジェクトにおいては、技術的に高度な専門知識が要求されます。

「AI技術を用いて経営課題を解決する」青写真を描いたとしても、技術的に実現可能でなければ意味がありません。
これは、課題から出発して、その課題を解決する技術を組み合わせるというパターンです。
反対に、技術を広く理解していれば、目の前の課題について複数の解決策を策定することができます。これは技術から出発して解決できる課題を見つけるパターンです。

データサイエンティストは分析ができることが先ずは求められます。データサイエンティストが顧客のビジネスを理解し(ドメイン知識を獲得し)、実際にシステム化する際の運用やシステム設計にまで明るくなれば、転職市場での価値はどんどん高まるでしょう。

データエンジニアの役割

AIを活用したシステムだけによらず、何かしらのシステムはデータを加工して指標を作成し、指標を基に人間の判断を促す、という動きをします。
AIを活用したシステムでは、使用するモデルの特性に合わせてデータを維持管理する必要があります。
オープンデータを活用するのであれば、強固なデータ収集の仕組みも必要でしょう。

データの設計、データを取り扱うシステムの設計は、システムの拡張性・汎用性を左右する非常に重要な業務です。
このように、AIを活用するシステムが滞りなく動くために必要なデータの加工、運用などを広く扱う職種です。

システムエンジニアの役割

AIモデルとユーザーを橋渡しするフロントエンド、AIモデルとデータストレージの動きを制御するバックエンドと活躍の舞台が大別して2つあります。

フロントエンドのシステムエンジニアは、顧客の業務への深い理解が求められます。ドメイン知識とコミュニケーション力を武器として、市場価値を高めていけます。

バックエンドのシステムエンジニアはAIモデルの挙動への深い理解が求められます。また大規模データの取り回しなど、ドメインによらない汎用的なスキルを身につけることができ、技術力を武器として市場価値を高めていくことができます。

演習1：AIを活用する視点（売上とコスト）

- AIシステムの事例を調査してください。
- 売上とコストの視点から導入の目的を説明してください。

演習2：AIを活用する視点（時間軸）

- AIシステムの事例を調査してください。
- 時間軸の視点から導入の目的を説明してください。

演習3：AIを活用する視点（競合他社との相対評価）

- AIシステムの事例を調査してください。
- 競合他社との相対評価の視点から導入の目的を説明してください。

演習4：AIを活用する視点（人の能力の拡張）

- AIシステムの事例を調査してください。
- 人の能力の拡張の視点から導入の目的を説明してください。

演習5：AIプロジェクトに関わる職種

- 皆さんが希望する職種は何でしょうか？本教材で取り扱った職種以外にも、様々な職種が存在します。
- 演習1～4で取り扱った事例から1つ取り上げ、皆さんが希望する職種はどのような成果を期待されているでしょうか？想像し、また皆さんの間で議論してみてください。

第2回：AIプロジェクトに関わる職種のスキルセットとキャリア構成

AIプロジェクトにおけるチームビルディング

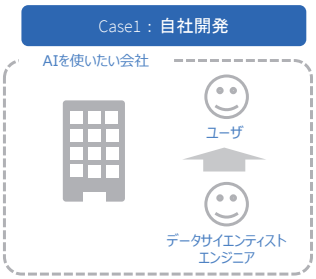
AI導入やデータ分析プロジェクトを遂行する場合、チームを組織することが一般的です。
プロジェクト遂行には様々なスキルが必要となりますし、作業ボリュームが1人でできる範囲ではないことがほとんどだからです。

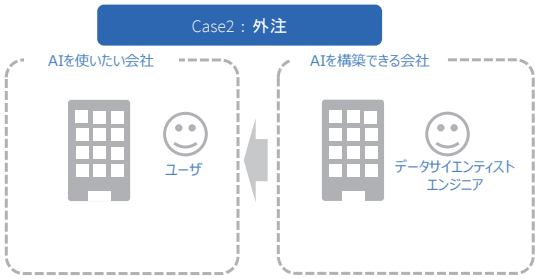
AI導入やデータ分析プロジェクトには、「できるかできないかやってみなければわからない」という類のプロジェクトが多数あります。
そのようなプロジェクトは、従来のシステム構築プロジェクトで採用されていたウォーターフォール型で実施されることはほとんどなく、「要件定義->分析->評価」を高速で繰り返すアジャイル型で実施されます。
チームメンバーがどのようなタスクを担っているのか、結果がどうだったのかをお互いに理解しあい、次の施策を決め、再び実行に移していきます。

そのような現場では、高いコミュニケーション能力とお互いの領域に対する理解・関心・敬意を持ちながらチームメンバーとして貢献することが求められます。

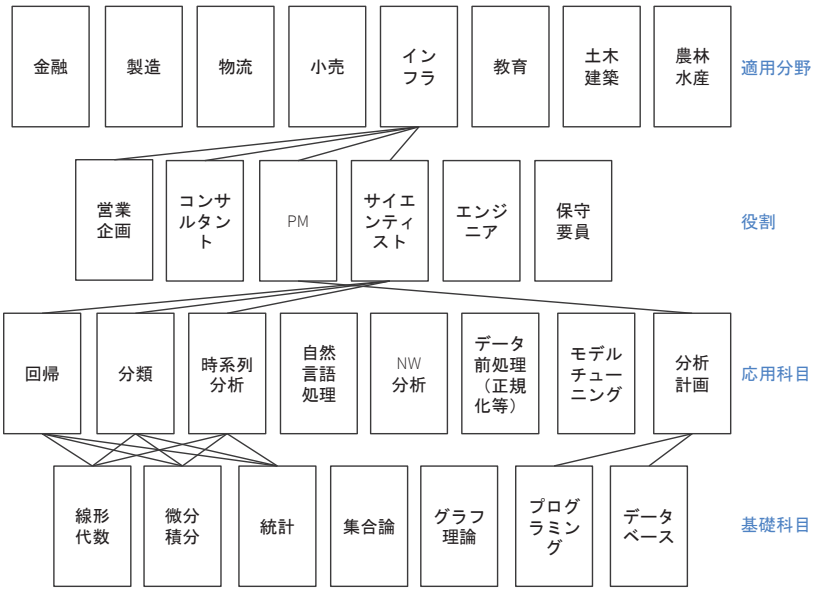
誰が実行するのか

- AIシステムの構築や分析プロジェクトを実施する場合、社内にAIに精通した人材がいれば自社開発・分析（Case1）を行います。社内にAIに精通した人材がいない場合、社外に発注（Case2）することになります。
- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に営業活動をすることになります。「AIを構築できる会社」の評判が良い場合、「AIを使いたい会社」からシステム構築を依頼されるケースもあります。





AIプロジェクトにおける役割・スキルセットの例

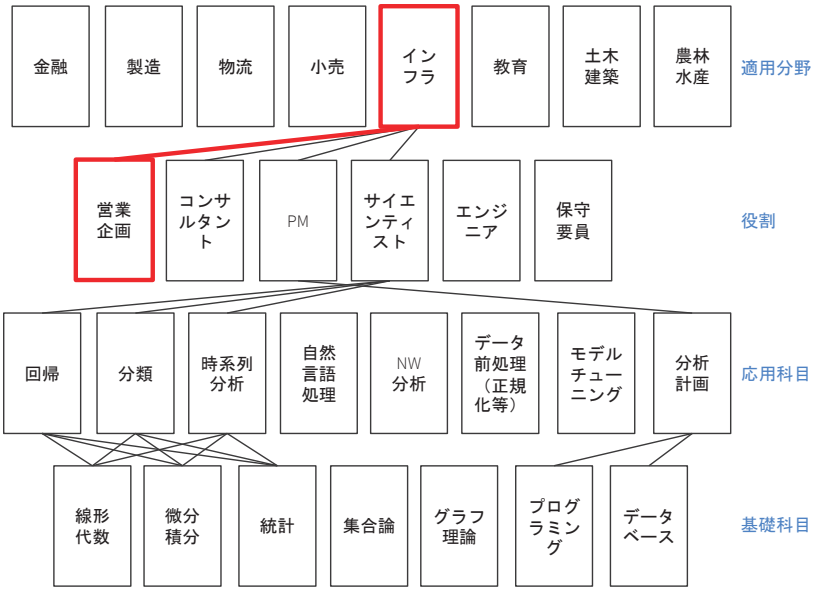


ある架空のプロジェクトを例として、プロジェクトメンバーの選定が行われる過程を説明していきます。

顧客はインフラを構築整備する企業であるとします（全ページのCase2に該当し、みなさんはAIを構築する会社の社員だとします）。
公共性の高いインフラですので、国や地方自治体の補助金で構築費用が賄われることがあります。一方、運用は地方自治体の独立採算が求められることがあります。

このようなビジネス環境で企業活動をしている顧客が、「自社の保有するデータを活用して全社的な利益率の向上を図りたい」と依頼してきたとします。

AIプロジェクトにおける役割・スキルセットの例

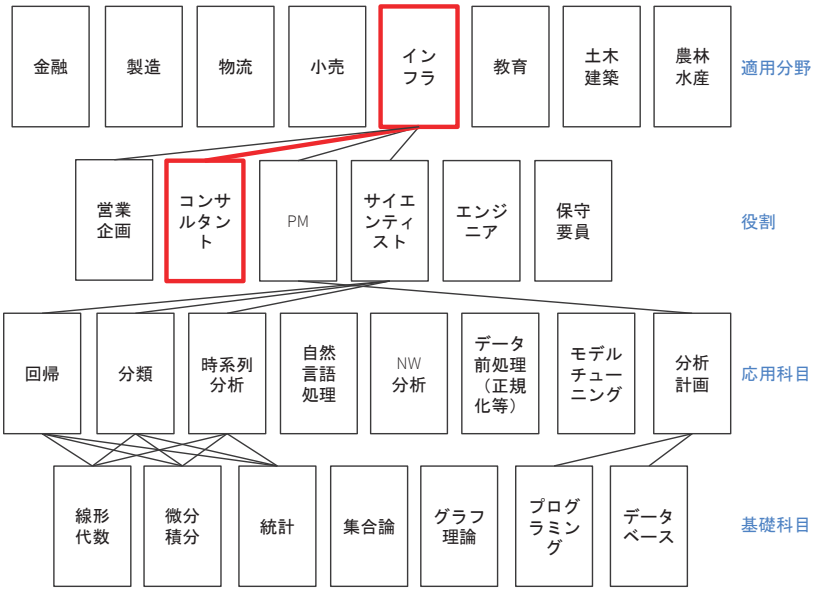


まず、営業企画をするメンバーの存在があります。
AI構築や分析を生業とする同業他社が増えている状況で、どのような分野に自社が進出すれば、参入障壁を高めて安定したポジションを築くことができるのか、常に自社の戦略を練り、業界にアンテナを張り巡らせています。

インフラは人間が経済活動をする限り存続するもので、もしインフラを生業とする顧客のビジネスパートナーとなれれば、自社にとっては安定に繋がります。

この戦略を実行に移した営業企画のメンバーが、この案件を受注しました。

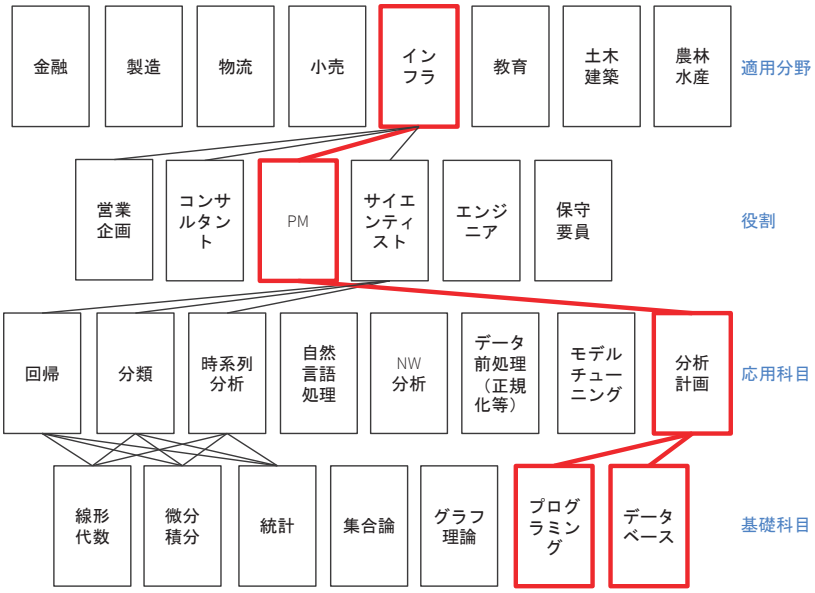
AIプロジェクトにおける役割・スキルセットの例



顧客が構築するのは公共性の高いインフラですので、国や地方自治体の補助金で構築費用が賄われることがあります。一方、運用は地方自治体の独立採算が求められることがあります。

このような特殊なビジネス環境で企業活動を行っている顧客のデータには、業界特有の事象が随所に現れてきます。
この特殊なデータが意味することを読み解き、顧客の利益率を向上させるためには、業界に精通しているコンサルタントのドメイン知識が必要となります。

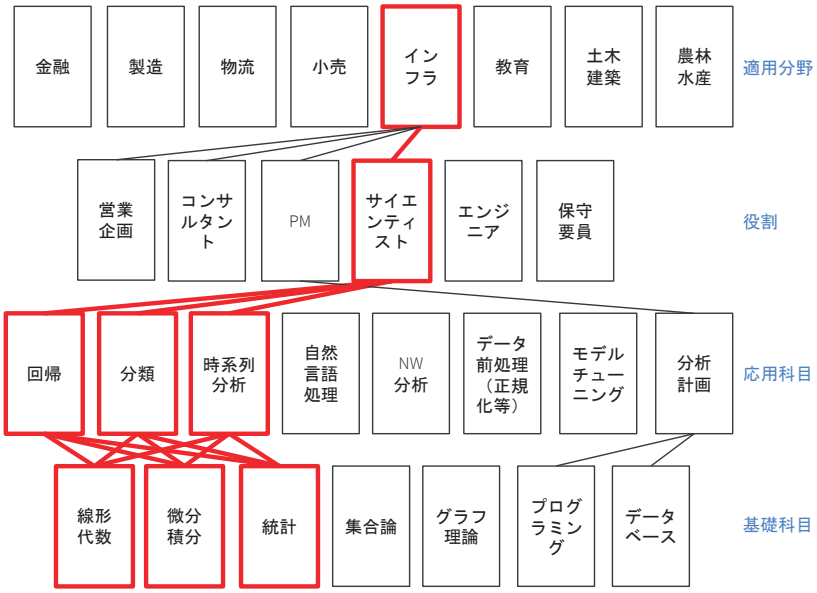
AIプロジェクトにおける役割・スキルセットの例



プロジェクト遂行するにはマネージャーが必要になります。
顧客の要望に答えるため、どのような分析をどの順番で実行すれば最も費用対効果が高いのか、分析の計画を立てます。

分析の計画は実現可能性があるものでなければならないので、プロジェクトマネージャーには技術に対する理解も求められます。

AIプロジェクトにおける役割・スキルセットの例



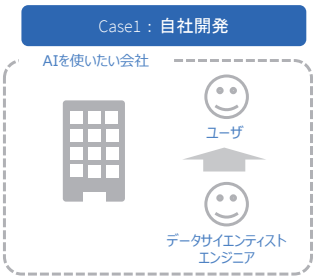
顧客のデータを実際に分析し、考察を加えて、利益率の向上を図るというシナリオを構築する必要があります。

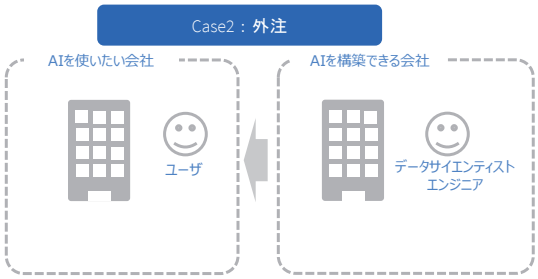
このようなプロジェクトは黒字/赤字になりやすい、このような遷移をするプロジェクトは黒字/赤字になりやすいなど、分類/回帰/時系列問題を組み合わせて分析する必要があります。

この様に、顧客の要望を実現するための分析技術を持つデータサイエンティストが必要となります。

AIシステム構築の対価

- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に対してAIシステム構築/分析サービスを実施し、対価をもらうことになります。
- プロジェクトに従事するメンバーの職種ごとに単価の相場があり、単価×稼働時間で対価を計算することが一般的です。
- 次ページ以降で、職種ごとの単価の目安値について説明していきます（※企業規模、外資/内資、企業のブランドによって金額は大きくばらつきますので、あくまで目安の値と捉えてください）。





職種ごとの単価の相場

パートナークラスのコンサルタントとなると、時間あたり5～10万円ほど請求することもあります。
マネージャークラスで時間あたり2～3万円程度、アソシエイトコンサルタントでも時間あたり1万円程度請求することがあります。

実際には、単価の高いパートナーやマネージャークラスが一つのプロジェクトにフルアサインすることはなく、月あたり500万円程度に抑えることが多いようです。
現場に張り付きで稼働するシニアコンサルタントやアソシエイトコンサルタントは、月当たり300～100万円程度となります。

職種ごとの単価の相場

チームをリードするデータサイエンティストは月当たり500～300万円ほど、メンバークラスのデータサイエンティストで月当たり200～100万円程度請求することになります。

データサイエンティストもコンサルタント同様に、単価の高い職位のメンバーは稼働を抑え、現場に張り付く月単価のメンバーを増やし、全体の金額を上げていくことが多いです。

※
企業のブランドによって、単価は大幅に上下します。
米国シリコンバレーに本社を構える某IT企業のデータサイエンティストとお話したことがあるのですが、その企業はデータサイエンティスト一人あたり月当たり1,000万円を顧客に請求するそうです。

職種ごとの単価の相場

データエンジニアやAI案件に携わるシステムエンジニアの相場は月当たり150～70万円程度となるようです。

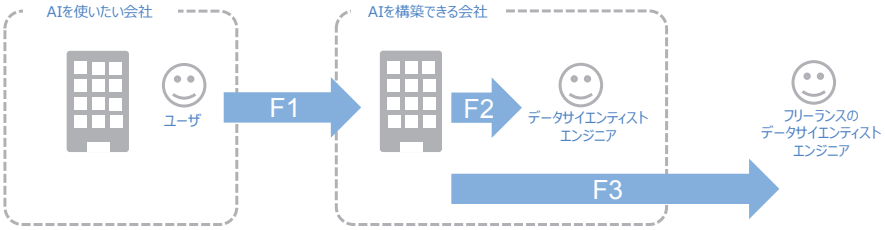
基幹系業務システムなどの大型案件でキャリアを築いてきたエンジニアがAIやデータ分析系キャリアに変更するときに、データエンジニアやシステムエンジニアは飛び込みやすい職種なので、自然とエンジニア人口が多くなります。供給が多くなり価格競争が起きやすい構図となっています。

商流について

AIやデータ分析に関するポジションは様々です。

- 「AIを使いたい会社」の社内にデータサイエンス部隊があり、そこで働く。
- 「AIを構築できる会社」で働く（前ページまでで扱ってきた単価はここに相当します）。
- フリーランスとして働く。

商流はシンプルな方が自分に入ってくる対価は多くなりますが、営業/事務作業コストも大きくなります。
対価が多くなるということは当然ながら成果物の品質をシビアに見られることになり、プレッシャーが大きくなります。
キャリアプランを立てるときは、自身の経験や人脈をしっかりと整理し、リスクを抑えてつつチャレンジしていくことが行くことが肝要でしょう。



演習1：営業・企画の役割

- あなたはあるシステム構築会社の営業企画職に就いているとします。どのような分野のプロジェクトを受注したいでしょうか？
- その分野の企業に営業に行く前に、業界の事前調査をすることにしました。あなたはどのような切り口で調査するでしょうか？

演習2：コンサルタントの役割

- 営業企画のチームメンバーから、一緒に顧客を往訪してくれと要請がありました。
- プロジェクトが受注間近なので、受注後に速やかにプロジェクトを開始できるように、顧客のことを知ってほしいとのことでした。
- 一週間後に、コンサルタントであるあなたは顧客を訪ねることになります。あなたはどのような準備をして臨むでしょうか？

演習3：データサイエンティストの役割

- データサイエンティストであるあなたは、コンサルタントとともに要件定義をしています。
- コンサルタントのチームメンバーは業務知識には明るいのですが、分析技術や数理ロジックには明るくありません。あなたは、どのようにチームメンバーと協力するでしょうか？

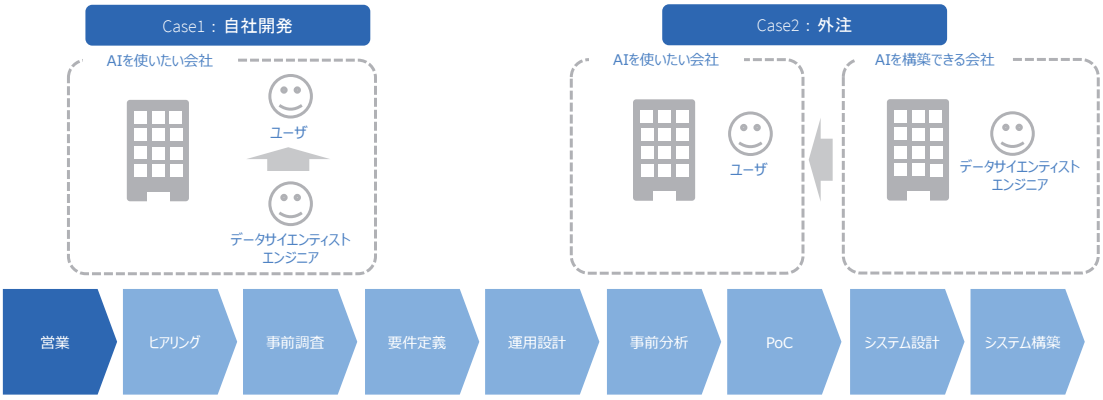
演習4：データエンジニアの役割

- データエンジニアであるあなたは、要件定義からプロジェクトに参画しました。
- 顧客がやりたいことを実現するために、データサイエンティストであるあなたのチームメンバーは、市場のトレンドが反映されたデータの分析をおこなっていました。
- あなたはデータサイエンティストから相談を受けました。取り扱うデータの特性上、トレンドが変化するととたんにモデルは精度が低下するそうです。モデルをリリースした後に精度が著しく低下した場合、すぐにモデルを切り戻すシステム構成にできないか、という相談でした。
- あなたが設計するシステムは、どのような機能が必要でしょうか？

第3回：AIプロジェクトのステップ

営業

- AIシステムを構築する場合、社内にAIを構築できる人材がいれば自社開発（Case1）を行います。社内にAIを構築できる人材がいない場合、社外に発注（Case2）することになります。
- Case2の場合、「AIを構築できる会社」は「AIを使いたい会社」に営業活動をすることになります。「AIを構築できる会社」の評判が良い場合、「AIを使いたい会社」からシステム構築を依頼されるケースもあります。



Case1：自社開発の事例

自社サービス開発を、自社のデータサイエンティスト/アナリストで実施しているケースです。
自社サービスで発生するデータを自由な発想で分析できることや、人月単価に縛られない価値を発揮できるなど、魅力的な環境だと思います。

参照：https://careers.mercari.com/jp/job-categories/analyst/

参照：https://linecorp.com/ja/career/position/588

Case2：他社のAIシステムを開発する事例

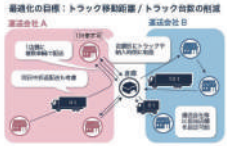
顧客のサービスやシステムを開発する事例です。
分野を横断してノウハウを習得することができます。いち早く有用な技術や方法論を習得し横展開することで、顧客から非常に重宝されます。

メガバンク初 AIによる住宅ローン事前審査　高度なスキルとAIの融合で、審査時間を約1日から最短15分へ短縮

株式会社三菱UFJ銀行様

金融

参照：https://jpn.nec.com/ai/case/index.html



業種　物流

活用技術　データ分析

属人化していた配車計画の立案業務を、数理最適化技術を用いて自動化し、ルートや配車台数などの最適化によりコスト削減を実現。

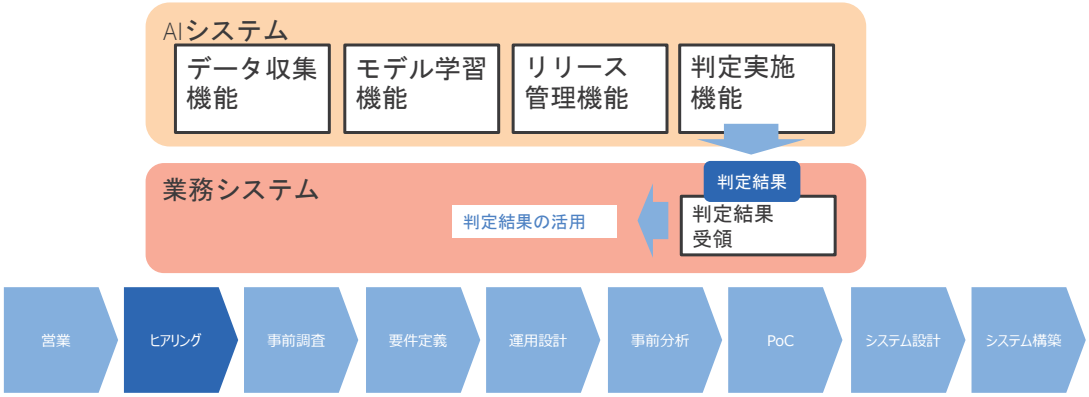
参照：https://ai.brainpad.co.jp/case-study/

AI導入で製品検査を効率化。生産ラインの変化・変動に即応できる画像認識システムとは？

参照：https://www.fujitsu.com/jp/solutions/business-technology/ai/ai-zinrai/customer-stories/

ヒアリング

- AIを使いたい会社・ユーザはAIを使って何を実現したいのか、ヒアリングを行います。
- 特に既存の業務システムとの連携を考えている場合は、業務システムの仕組み上実現が難しい場合や費用が大きくなる場合もあるため、導入コストと得られるメリットの双方を勘案する必要があります。
- AIを導入しなくてもユーザのやりたいことが実現できると判断する場合や、コストが大きすぎて導入を避けた方や良い場合は、プロジェクト自体の中断を決断することになります。



事前調査

- AI導入にメリットがあると判断した場合、データ・技術の事前調査へと進めます。
- 機械学習などを用いてAIモデルを作成するためには、データの量と質がモデルを構築するに足るかどうかを判断する必要があります。

- データの量と質が基準を満たしている場合でも個人情報が含まれていたり、コンプライアンス（法令遵守）の視点からデータの使用が困難であったりする場合があります。コストを投入する前に様々な角度からプロジェクトの実現性を検証する必要があります。

要件定義

- まずはシステムが満たすべき条件を決定します。例えばモデルのインプットデータの項目と量、モデルのアウトプットの項目、モデルの精度などです。
- モデル以外にも蓄積するデータの量と、蓄積の速度・期間などについて決定します。

- システム以外にも、サービスとして満たすべき条件を決定します。例えばユーザーからのリクエストに対して何秒以内に応答を返さないといけないか、一日何回の使用に耐えなければならないか、システム運用のコストと売上の損益分岐をどこに定めるかなどです。

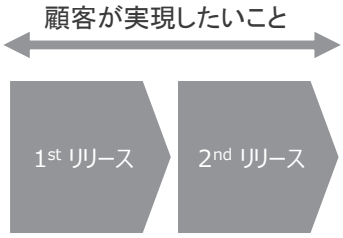
- 実務においては後述する運用設計・事前分析・PoCを数回繰り返して要件を決定していきます（実際にやってみて発覚することがあるため、このような繰り返しのやり方が必要となります）。

「スコープを切る」

顧客が実現したいことをシステムなどで具現化する際に、あえて段階的に実施することがあります。

仕事を受ける側からすると、一度に受注したほうが売上が大きくなるという誘惑に駆られますが、失敗したときのダメージが大きくなります。
一度に全てのことを実現するには複雑すぎてリスクが見通せない場合や、あえて踊り場を設けて実験を繰り返しながら着実に積み上げる場合など、スコープを切る理由は様々です。

変化に対応しようという意欲が高くなるほど「あれもこれも」と前のめりになってしまいます。多くの事例を経験し、きちんとリスクを顧客に伝えることができる人材は、どの現場でも重宝されます。



運用設計

- AIモデルを実サービスとして展開し続けるためには、運用が必要になります。
- 増え続けるデータの蓄積、モデルの更新、不具合への対応にどれだけの人・機材・お金・時間を投入するのかを決定します。投入するこれらのコストを勘案し、ユーザに請求するサービス料の設定が必要となります。

- ユーザの予算が決まっていて多額の投資をできない場合、システム機能の削減や運用フローの簡素化を実施します。

「運用設計」はいつ実施するのが適切か？

本資料では、「運用設計」は「事前分析」や「PoC」の前のステップとして記載されています。

「やってみなければできるかどうかわからない」というAI案件は、PoC（概念実証）という実現可能性を検証するステップを置くことがあります。

PoCでは素晴らしいモデルが構築され良い結果が出たとしても、いざ実システムの設計や運用を考える段階になって、「実現するのは難しい」という結論になってしまうことが少なくありません。
これはPoCのために特別に仕立てたデータを使用していたり、個人情報保護やセキュリティの問題をPoCの最中だけ特別にクリアできる条件で実施していたりと、本来の条件ではない条件で実施することがあるからです。

運用設計を早い段階で実施することで、「本当に実現できるのか？」ということを早めに熟慮せざるを得なくなります。
早い段階でだめだとわかれば、その後投下する予定だったヒト・モノ・カネを節約することができます。

事前分析

- モデル作成に使用するデータの詳細な調査を実施します。
- モデルの目的変数、説明変数の統計値を算出したり、目的変数と説明変数間で相関があるのかなどの検証を実施します。

- 本プロセスにおいてデータの大きな偏りが確認されたり、また説明変数と目的変数間に全く相関が確認されない場合、機械学習を導入しても効果が見られないと判断してプロジェクトを中断することもあります。このステップはプロジェクトの「勝ち目」を判断する重要なステップです。

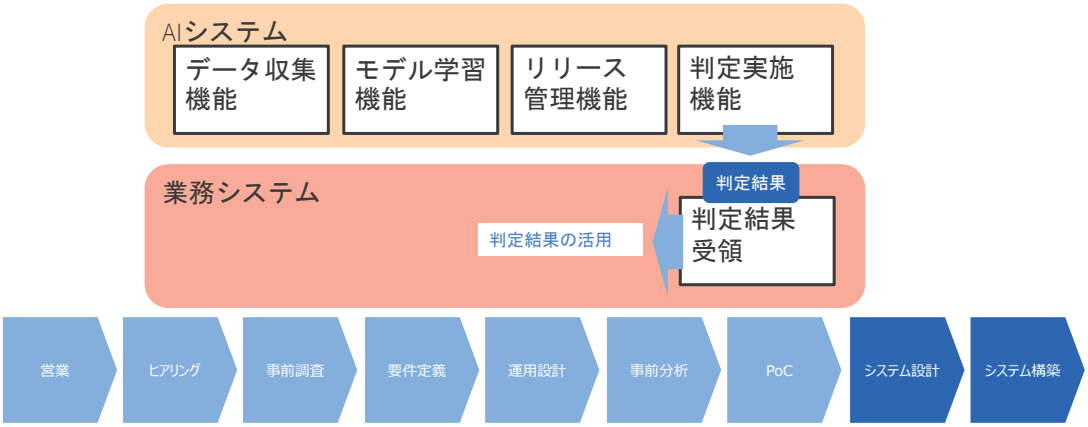
PoC（Proof of Concept：概念実証）

- AIモデルが本当に実現できるのか、実際に試してみるステップです。
- データ加工、機械学習アルゴリズムの試行、精度検証をいかに正確に積み上げていけるかがプロジェクト成否の分かれ目となります。機械学習の実施が効率化できるライブラリの使用や、独自スクリプトの構築が欠かせません。

- このステップで大事なのはモデルの精度向上ではありますが、要件定義で決定した事項を満たせているのか、ということを常にチェックしなければなりません。例えば「精度の向上を優先しすぎるあまりDeep Neural Networkをアルゴリズムとして選択してしまい、モデルの説明性が低下する。」という事態を避けなければなりません。

システム設計・システム構築

- モデルを作成した後は、データ収集からモデル判定結果の提供方法までを含めたシステム設計が必要となります。
- 設計後は実際にシステムを構築していきます・



演習1：AIシステム構築の主体

- 自社サービス/システムを自社のデータサイエンティスト部隊で開発している事例を調査してください。
- 顧客のサービス/システムを構築する事例を調査してください。
- あなたが参画するとしたら、どちらのケースが良いでしょうか。その理由について整理してください。

第5回：データ収集

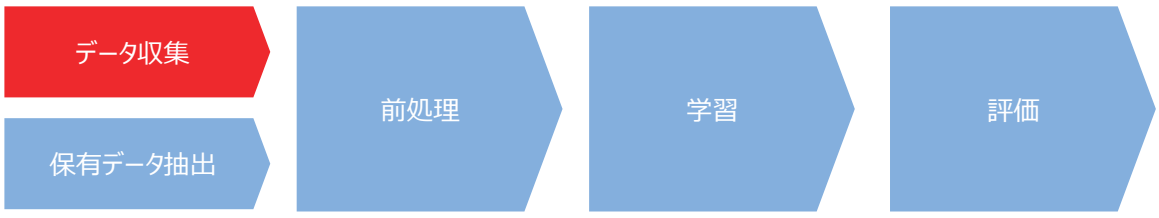
アジェンダ

- データ収集が必要となる場面
- データ収集先の事例
- HTML/PDFからのデータ抽出

データ収集が必要となる場面

データ収集が必要となる場面

- データ分析、モデル構築を行うためにはデータが必要です。すでに保有しているデータで目的が達成できる場合、データの収集は必要ではありません。
- 保有データだけでは足りない場合や、保有データだけでも一通り分析やモデリングはできるのですが、精度を更に向上させたい場合など、新規にデータを収集することになります。



モデル作成のステップ

データ収集が必要となる場面

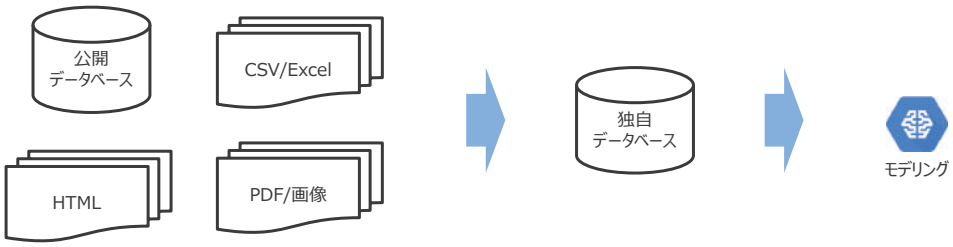
独自にデータを収集しなければならない事例には、下記の場面があります。

- 特定の形状の物体を画像認識させたいが、公開されているモデルではその形状が学習されていない。
- これまで新出していなかったエリアに新規出店したい。市場規模や地域特性を分析したいが、新規エリアなのでデータがない。
- 公開されているモデルで代用できると思ったが、ライセンスが商用利用不可だった。独自にモデルを作成するしかない。

モデル作成のステップ

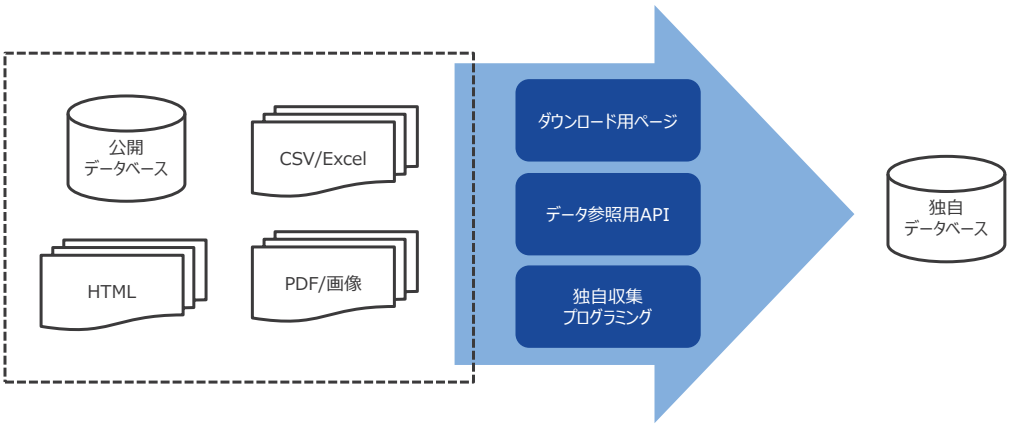
収集するデータの形式

- 収集対象のデータ形式は多岐に渡ります。
- 既に使いやすい形式（データベース、CSVなど）になっていることは稀で、モデリングに使いたい箇所のみをうまく抽出する必要があります。



収集するデータの形式

- インターネットを経由してデータを収集する場合、ダウンロード専用のページやAPIが公開されていれば、あまりコストを掛けずにデータを収集することができます。
- 収集の手段が用意されていなければ、独自に収集の仕組みを構築する必要があります。



データ収集先の事例

データ収集フロー

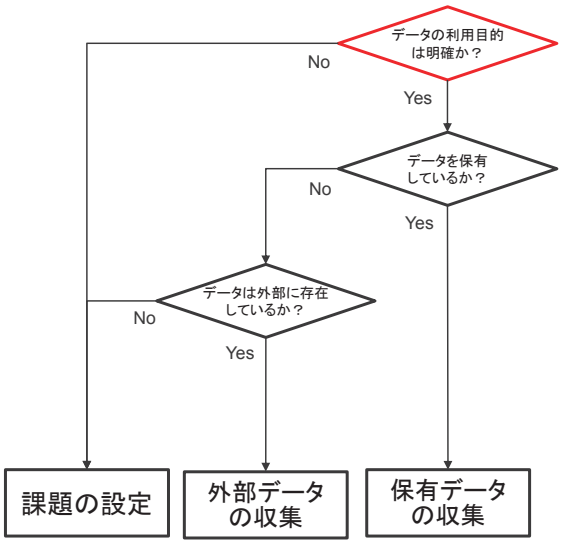
データ収集にはコストが掛かるため、予め計画を立てて実行することがおすすめです。

収集するデータを何に活用するのか、目的を明確化させておくことが重要です。
可能であれば、

- モデルのXXXの説明変数を強化する、
- 保有データのXXX項目と内部結合する、
- 収集データを加工してYYYの意味を持つ項目を作成する、

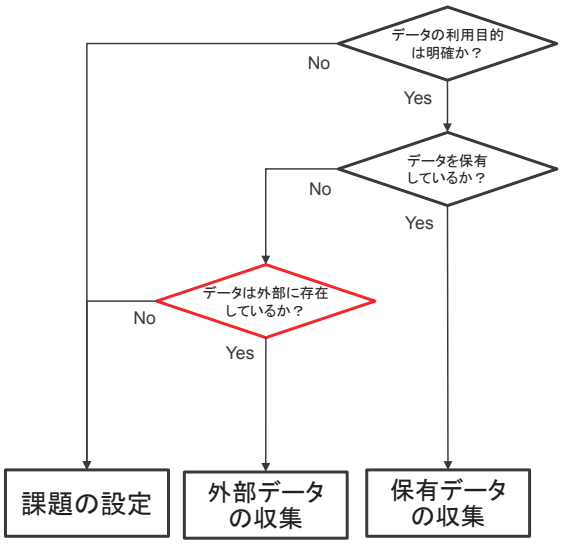
などより具体的な設定がよいです。

項目だけでなく、データ量もチェックしなければなりません。「トレンドを把握するために時系列分析したいのに、ある時間断面のデータしか存在しない」となると、目的を達成することはできません。



データ収集フロー

データを保有していない場合、外部データを調達することになります。
データがCSVやExcelで公開されていれば分析に活用しやすいのですが、WordやPDF、HTMLに埋め込まれた形で公開されている場合もあります。



オープンデータの事例：RESAS

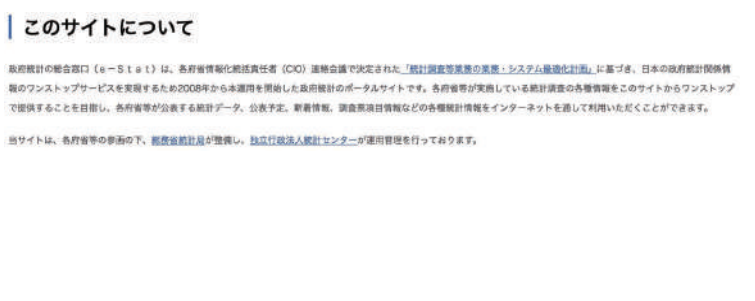
経済産業省と内閣官房（まち・ひと・しごと創生本部事務局）が提供している地域経済分析システムです。
APIでデータにアクセスすることができます。

参照：https://resas.go.jp/#/13/13101

オープンデータの事例：e-Stat

各府省等の参画の下、総務省統計局が整備し、独立行政法人統計センターが運用管理を行っている政府統計のポータルサイトです。
ExcelやPDFでのデータダウンロード、APIによるデータへのアクセスが可能です。

参照：https://www.e-stat.go.jp/



参照：https://www.e-stat.go.jp/about

オープンデータの事例：その他の事例

地方公共団体や企業もオープンデータを公開しています。
興味のある方は検索してみてください。

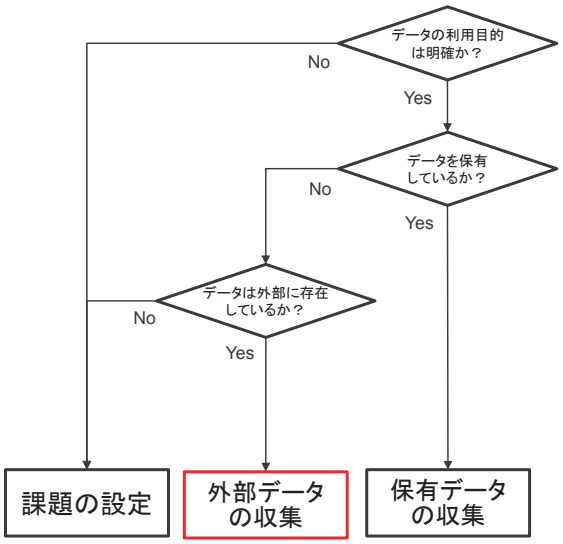
No.	事例	概要・テーマ	事業者等
1	会津若松市消火栓マップ	スマートシティ会津若松の実現へ	Code for Aizu
2	アグリノート	農業×ICTを支えるオープンデータ	ウォーターセル株式会社
3	イーグルバス	運用状況の見える化へのチャレンジ	イーグルバス株式会社
4	カーリル	日本だからできた図書館システム	株式会社カーリル
5	家計簿・会計アプリZaim	公共データでサービスを格上げ	株式会社Zaim
6	かなざわ育なび.net	行政データを集約してひとり一人にあわせてリアルタイムに反映	横浜市 金沢区
7	花粉くん	オープンデータを"可愛く"使う	株式会社博報堂アイ・スタジオ
8	ココゆれ	オープンデータで付加価値を	大和ハウス工業株式会社
9	5374(ゴミナシ).jp	コードで地域課題を解決する	Code for Kanazawa
10	さっぽろ保育園マップ	分散化したデータを一元的かつ容易に閲覧できる	Code for Sapporo（パパママまっぷチーム）

参照：https://cio.go.jp/opendata100

HTML/PDFからのデータ抽出

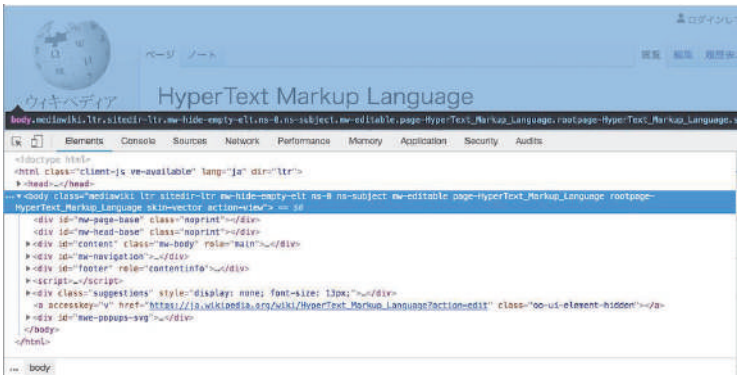
HTML/PDFからのデータ抽出

収集したデータがHTMLやPDFの場合、欲しい情報だけを抽出することが必要です。



HTMLのスクレイピング

- HTMLとは、Hyper Text Markup Language（ハイパーテキスト・マークアップ・ランゲージ）の略で、Webページを作るための最も基本的なマークアップ言語のひとつです。
- 普段、私たちがブラウザで観ているWebページのほとんどが、HTMLで作られています。



Google ChromeからWikipediaのページをHTMLで表示した例

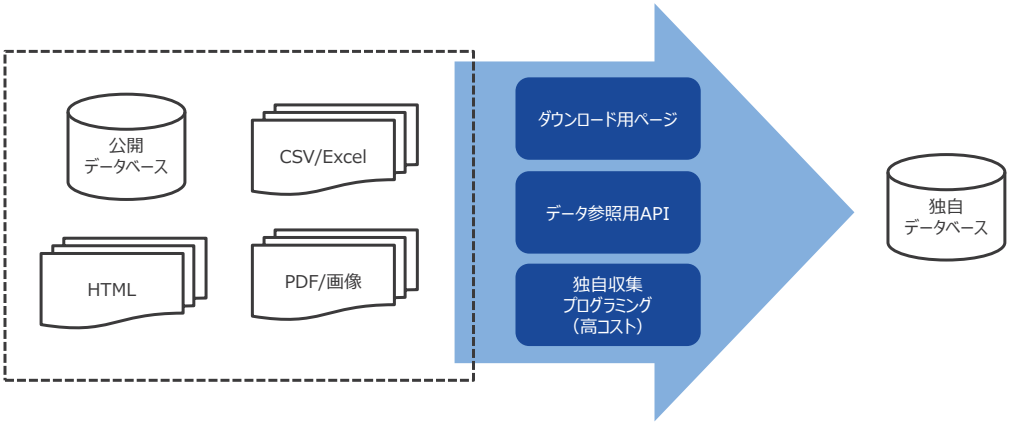
HTMLのスクレイピング

- 例えばWikipediaのページ(https://ja.wikipedia.org/wiki/HyperText_Markup_Language)の見出しから「HyperText Markup Language」という文字列を抽出するためには、HTMLのヘッダ部分にある<h1>タグで挟まれている部分を見つけ出して、切り出す必要があります。このような操作をスクレイピングといいます。

Google ChromeからWikipediaのページをHTMLで表示した例

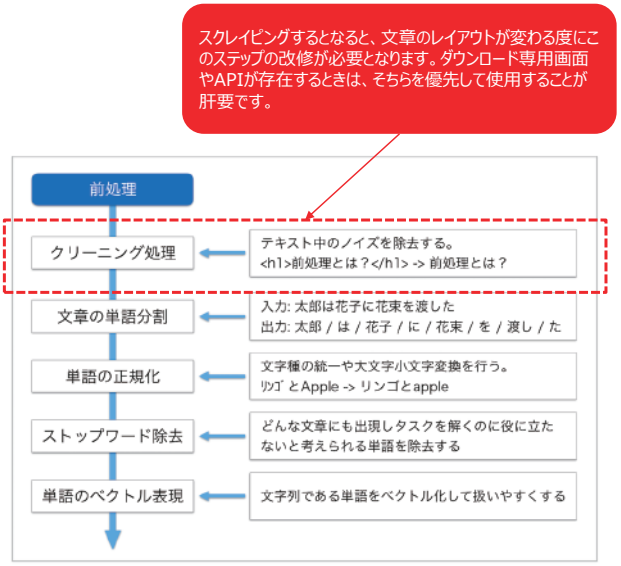
収集するデータの形式

- 前ページではWikipediaのページをスクレイピングする話を記載しました。HTMLを手元に落とし、中身を確認してタグを除去してと、非常に手間がかかることがわかります。
- 実はWikipediにはAPIが存在し、はるかに低コストで文字列を取得することができます。



データ収集の戦略

- HTMLをスクレイピングするのとAPIを利用するのでは、データの取得コストが大きく違うことを学習しました。
- 機械学習モデルを作成するためには大量のデータを必要とするため、データ収集の仕組みはそれ自体で1つのシステムとなるほどの規模です。
- 取得対象とするデータは何なのか、どのような形式で取得できるのか、予め計画を立ててプロジェクトを進めなければ、データ収集コストが肥大するのを避けられなくなってしまいます。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

データ収集の戦略

- データの取得にコストが掛かるということを学習しました。また、機械学習モデルを構築するためには大量のデータが必要だということも学習しました。
- データを使う側の立場としては、APIを駆使して低コストでデータを収集することが肝要ですが、APIが存在しないデータをあえてスクレイピングで取得することがあります。

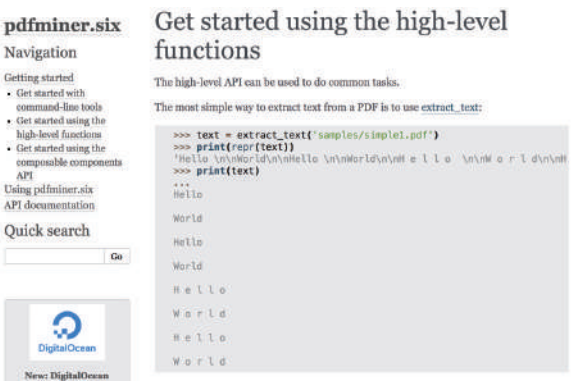
- 繰り返しになりますが、機械学習モデルの構築にはデータが必要です。そのデータは、簡単に取得できないものでかつ需要があるほど高価になります。モデリングする企業に対してデータを売却することをビジネスとする企業が存在するほど、ビジネスとしてはホットな分野なのです。

PDFからのデータ抽出

PDFからテキストを抽出するライブラリがあります。
下の絵はpdfminer.sixというPythonのライブラリです。

PDFはExcelのように列が揃っているわけではないため、抽出がスムーズにできるかは試してみてわかることが多いです。

まずテキスト列を抽出し、規則性を見つけて必要な箇所のみ抽出するという作業となります。



参照：https://pdfminersix.readthedocs.io/en/latest/tutorials/highlevel.html

演習

演習1：データ収集

- オープンデータを活用してサービスを運用している企業とその仕組について調べてみてください。

第6回：音声認識

アジェンダ

- 音声認識の仕組み
- 音声認識の事例

音声認識の仕組み

音声認識とは

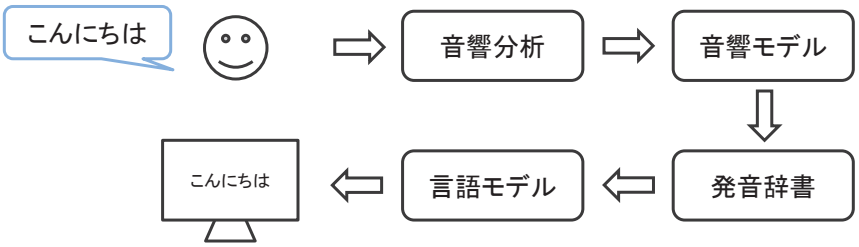
[Wikipediaより]
音声認識（おんせいにんしき、英: speech recognition）とは、人間の声などをコンピューターに認識させることであり、話し言葉を文字列に変換したり、あるいは音声の特徴をとらえて声を出している人を識別する機能を指す。

音声認識の処理プロセス

音声認識は大別して以下の4つのプロセスで構成されます。

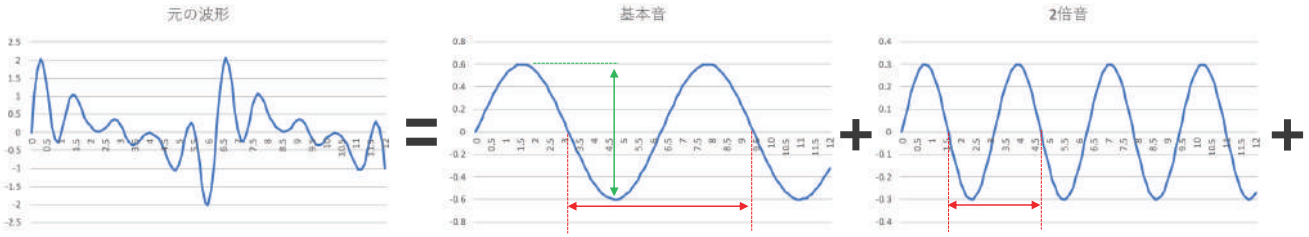
- 音響分析
- 音響モデル
- 発音辞書
- 言語モデル

DNN（Deep Neural Network）が登場してからは、必ずしも4つのプロセスを別々に実行するのではなく、一つのプロセスとして実装されることが多くなっています。



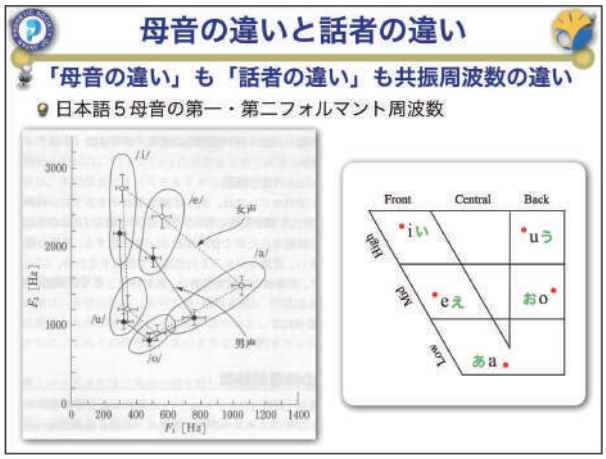
音響分析

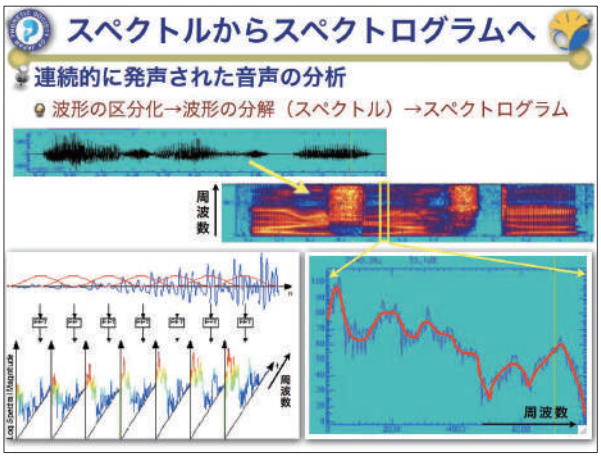
入力された音声データ（空気の振動・波）を加工し、音響モデルで扱える形に変換します。
例えば、音の強弱や周波数を抽出します。



音響分析

入力された音声データ（空気の振動・波）を加工し、音響モデルで扱える形に変換します。
共振による母音の特定、周波数の時系列変動などを特徴量として抽出することもされています。

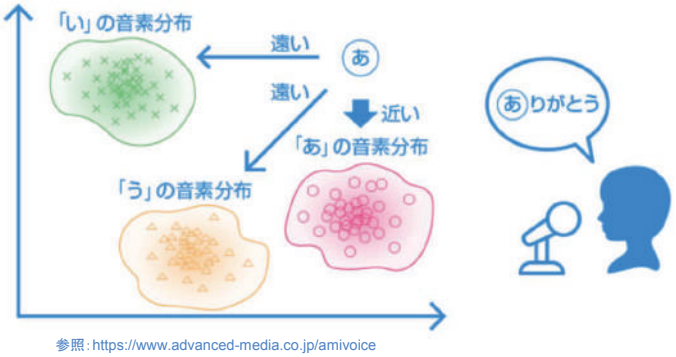




参照：日本音声学会音声学普及委員会
https://www.gavo.t.u-tokyo.ac.jp/~mine/japanese/nlp+slp/lecture-02.pdf

音響モデル

音響分析で音声データから抽出した特徴を基に、発音された音がどの音素に近いのかを特定します。



参照：https://www.advanced-media.co.jp/amivoice

発音辞書

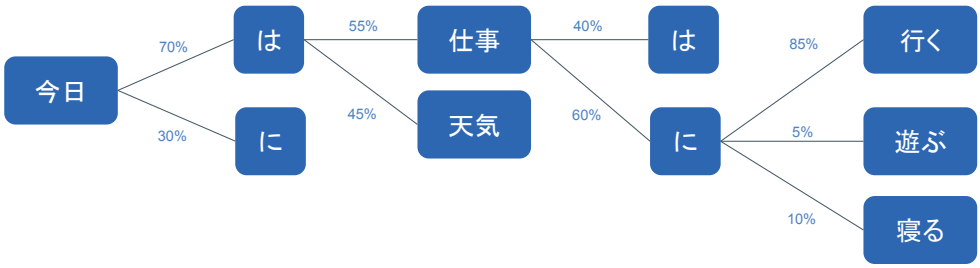
言語モデルの単語と音響モデルを結びつけるために発音辞書が使われます。
音声の最小単位の「音素」ごとにモデル化したデータベースから音の組み合わせを抽出し、「単語」として認識させます。

単語	読み	音素列
哀れ	あわれ	a w a r e
哀願	あいがん	a i g a N
愛	あい	a i
遭っ	あっ	a t t a

参照：https://www.advanced-media.co.jp/amivoice

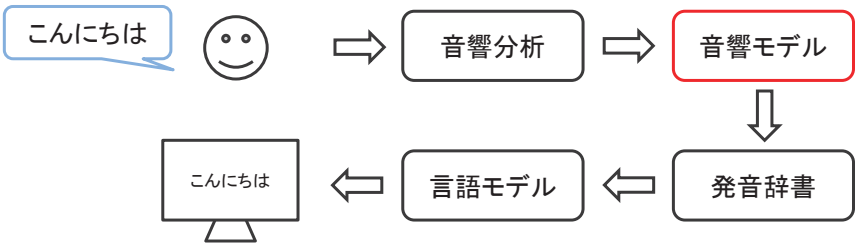
言語モデル

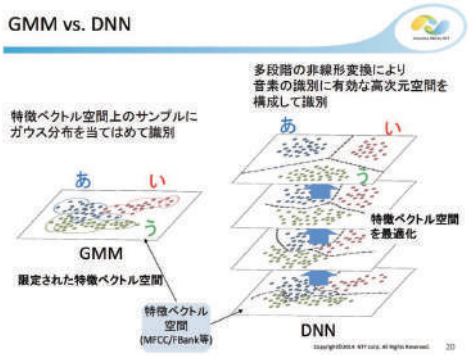
隠れマルコフモデル（Hidden Markov Model）は言語モデルとしてよく利用されます。
HMMはある文字列に続く直後の文字の出現しやすさ（出現確率）を計算します。



DNNの活用

音響分析で作成した特徴量から音素を推定するプロセスにおいて従来はガウス混合分布（Gaussian Mixture Model; GMM）により音素を推定していましたが、近年はDNNが活用されるようになってます。

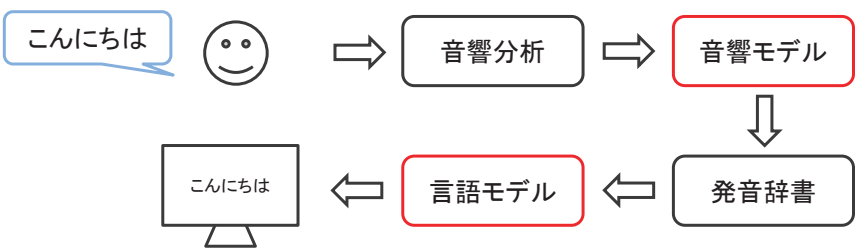




参照：http://asj-fresh.acoustics.jp/wordpress/wp-content/uploads/2015/12/2015f_beginners_fujimoto.pdf

DNNの活用

音素の推定のみでなく、言語モデルで使用されているHMMと組み合わせてDNNが活用されることもあります。ある時刻のHMMの最尤隠れ状態を正解ラベルとし、入力を音素特徴量とし、DNNをトレーニングします。



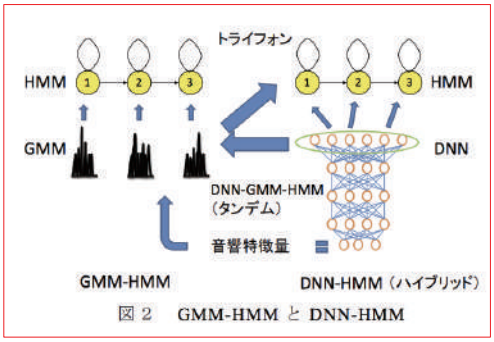


図２　GMM-HMM と DNN-HMM

参照：http://sap.ist.i.kyoto-u.ac.jp/members/kawahara/paper/KAW-prmu15-12.pdf

HMMとDNNについて学習したい方へ

HMMとDNNについて更に学習したい方向けに、参考となりそうなURLを記載します。

【HMM】
https://www.gavo.t.u-tokyo.ac.jp/~mine/japanese/nlp+slp/IPSJ-MGN451003.pdf
http://kento1109.hatenablog.com/entry/2017/12/15/160315
https://mieruca-ai.com/ai/markov_model_hmm/

【HMM+DNN】
https://www.jstage.jst.go.jp/article/jasj/73/1/73_31/_pdf
http://asj-fresh.acoustics.jp/wordpress/wp-content/uploads/2015/12/2015f_beginners_fujimoto.pdf
http://sap.ist.i.kyoto-u.ac.jp/members/kawahara/paper/KAW-prmu15-12.pdf

音声認識の事例

音声認識API

音声認識技術を活用すると音声をテキストに変換することができるため、
テキスト化した情報を基に様々なサービスに活用することができます。

参照：https://www.advanced-media.co.jp/amivoice

自然言語処理との連携

テキスト化したデータを基に自然言語処理を実行し、対話を促進する仕組みもあります。
構文解析や意味解析も実行されているため、発話に含まれるキーワード（場所、天気、時間など）も認識したうえでの
サービスも設計できるようです。

参照：https://www.kiku-hana.jp/

演習

演習1：音声認識技術の活用事例

- 音声認識技術を活用したサービスについて、サービス内容と技術的な仕組みについて調べてみてください。

第7回：画像認識

アジェンダ

- 画像認識の仕組み
- 画像の加工
- 画像認識の事例

画像認識の仕組み

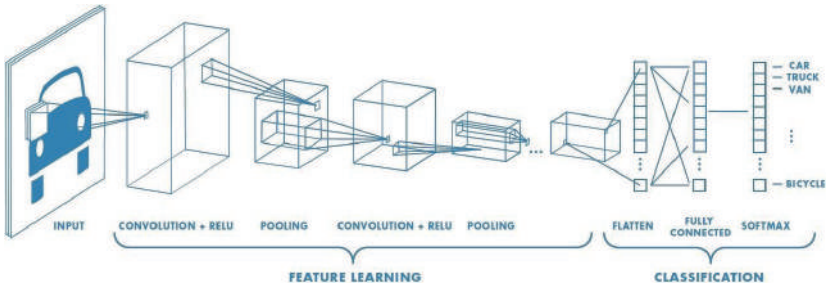
画像認識とは

画像認識とは、画像に含まれる物体をコンピューターや機械などで識別できるようにする技術のことです。
画像から色や形といった特徴を抽出し、その特徴を基に画像を認識できるようにするパターン認識技術のひとつです。

DNN（Deep Neural Network）が登場してからは、画像認識にDNNが活用されることが非常に多くなっています。

1. 対象物の分類問題

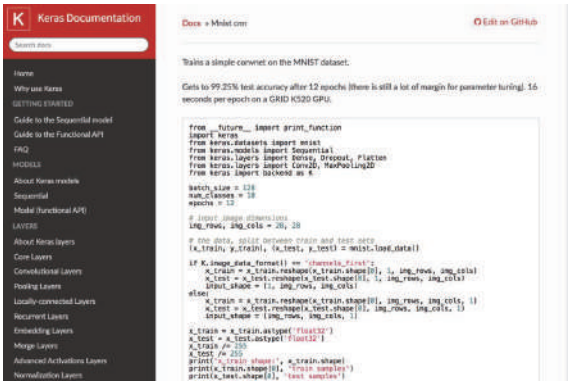
画像認識で取り扱う問題の1つとして、画像に判定したい対象が1つ写っているのみで、その対象物が何なのかを分類するという問題があります。



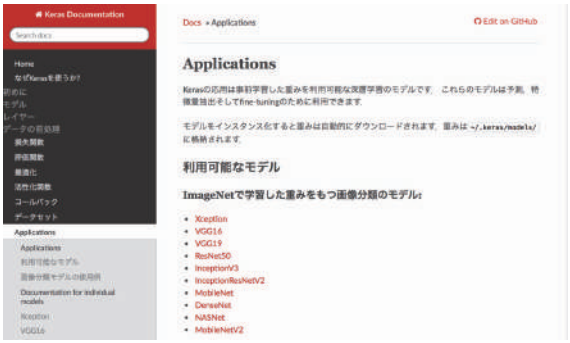
参照：https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

1. 対象物の分類問題

分類問題を実行するCNN（Convolutional Neural Networks）のソースや、学習済モデルは簡単に利活用することができます。



参照： https://keras.io/examples/mnist_cnn/

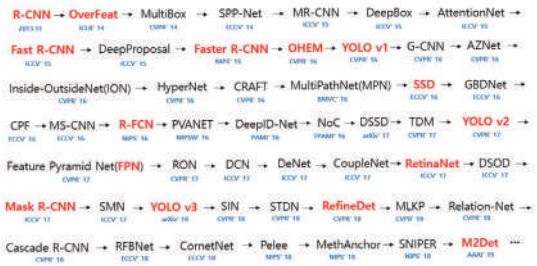


参照： https://keras.io/ja/applications/

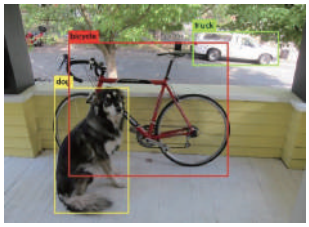
2. 対象物の抽出

画像に判定したい対象が複数写っている場合に、その複数の対象物をそれぞれ抽出し、かつ何なのかを分類する「一般物体認識」という問題があります。

CNNを発展させた様々なアルゴリズムが考案され、公開されています。



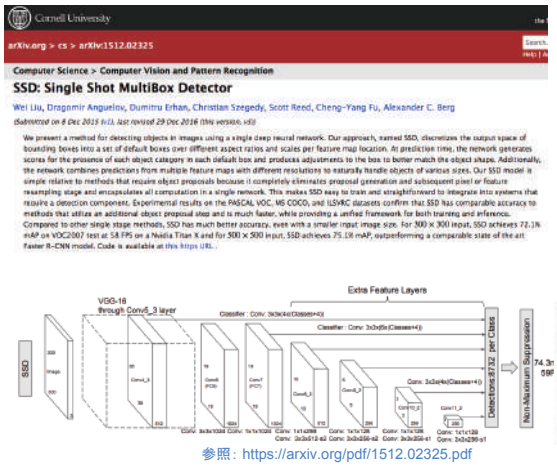
参照：https://github.com/hoya012/deep_learning_object_detection



参照：https://pjreddie.com/darknet/yolo/

2. 対象物の抽出：SSD（Single Shot MultiBox Detector）

一般物体認識を実行するアルゴリズムの例として、SSDがあります。
SSDは、判定領域のサイズを変えながらクラス分類をするタスクを繰り返し、モデルの重みを最適化します。



参照：https://arxiv.org/pdf/1512.02325.pdf

2. 対象物の抽出：U-Net

CNNをAuto-Encoderのように使用するアルゴリズムです。
入力データのマスク箇所をデコードできるように、ネットワークがチューニングされます。
マスク箇所の境界線情報を喪失しないように、デコード箇所でエンコード時の情報をコンカチします。

```
def create_unet(self):
    s = Input(self.sz)
    c1 = Conv2D(8, 3, activation='relu', padding='same') (s)
    c1 = Conv2D(8, 3, activation='relu', padding='same') (c1)
    p1 = MaxPooling2D() (c1)
    c2 = Conv2D(16, 3, activation='relu', padding='same') (p1)
    c2 = Conv2D(16, 3, activation='relu', padding='same') (c2)
    p2 = MaxPooling2D() (c2)
    c3 = Conv2D(32, 3, activation='relu', padding='same') (p2)
    c3 = Conv2D(32, 3, activation='relu', padding='same') (c3)
    p3 = MaxPooling2D() (c3)
    c4 = Conv2D(64, 3, activation='relu', padding='same') (p3)
    c4 = Conv2D(64, 3, activation='relu', padding='same') (c4)
    p4 = MaxPooling2D() (c4)
    c5 = Conv2D(128, 3, activation='relu', padding='same') (p4)
    c5 = Conv2D(128, 3, activation='relu', padding='same') (c5)
    u6 = Conv2DTranspose(64, 2, strides=(2, 2), padding='same') (c5)
    u6 = Concatenate(axis=3)([u6, c4])
    c6 = Conv2D(64, 3, activation='relu', padding='same') (u6)
    c6 = Conv2D(64, 3, activation='relu', padding='same') (c6)
    u7 = Conv2DTranspose(32, 2, strides=(2, 2), padding='same') (c6)
    u7 = Concatenate(axis=3)([u7, c3])
    c7 = Conv2D(32, 3, activation='relu', padding='same') (u7)
    c7 = Conv2D(32, 3, activation='relu', padding='same') (c7)
    u8 = Conv2DTranspose(16, 2, strides=(2, 2), padding='same') (c7)
    u8 = Concatenate(axis=3)([u8, c2])
    c8 = Conv2D(16, 3, activation='relu', padding='same') (u8)
    c8 = Conv2D(16, 3, activation='relu', padding='same') (c8)
    u9 = Conv2DTranspose(8, 2, strides=(2, 2), padding='same') (c8)
    u9 = Concatenate(axis=3)([u9, c1])
    c9 = Conv2D(8, 3, activation='relu', padding='same') (u9)
    c9 = Conv2D(8, 3, activation='relu', padding='same') (c9)
    outputs = Conv2D(1, 1, activation='sigmoid') (c9)
    self.model = Model(inputs=[s], outputs=[outputs])
```

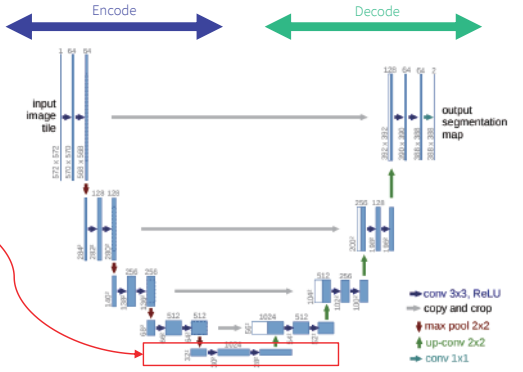


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

画像参照元：https://arxiv.org/pdf/1505.04597.pdf

画像の加工

特徴量エンジニアリング

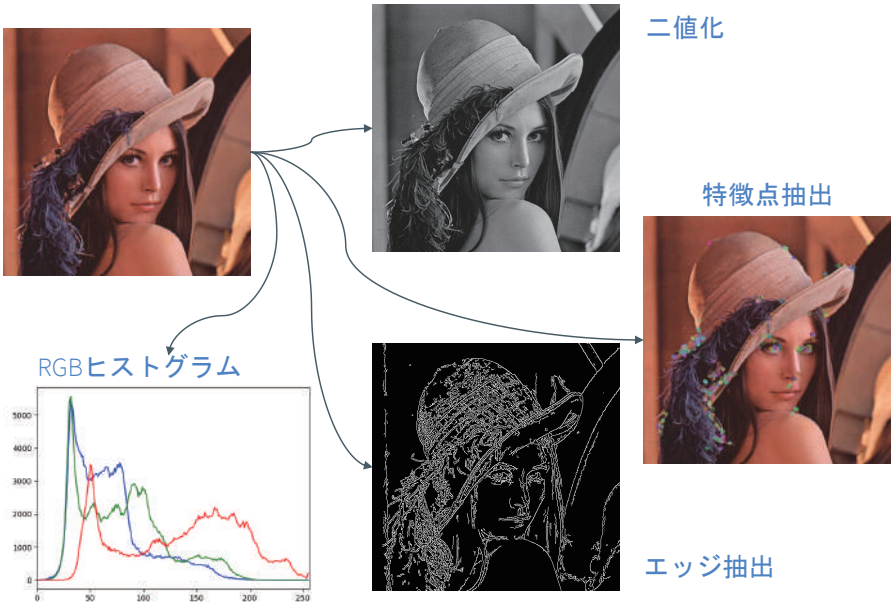
前ページまでで、CNNなどで画像に写っている物体の判定をすることを学習してきました。
学習器の精度を向上させるためには、事前に画像に何かしらの加工処理を加えることがあります。

加工処理を実行するライブラリとして、以下のようなものがあります。

- OpenCV（C/C++、Java、Python）
- skimage（Python）

OpenCV：画像加工とアルゴリズム

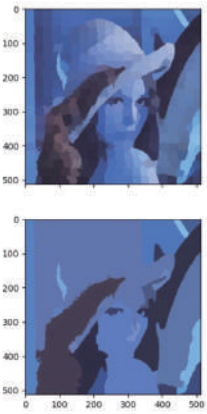
OpenCVでは右の画像に記載しているような加工ができます。



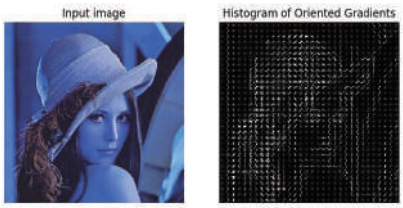
skimage：画像加工とアルゴリズム

skimageでは右の画像に記載しているような加工ができます。

RAG
色の違いで切り分ける



HoG特徴量抽出
輝度の勾配をヒストグラム化



画像認識の事例

画像認識AIレジ

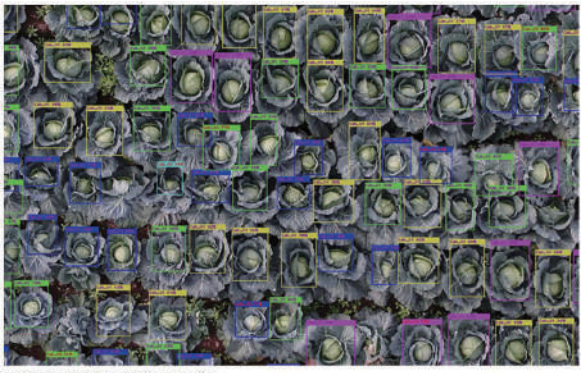
パンを認識し、レジ打ち作業を簡略化した事例が紹介されています。
https://ledge.ai/bakery-scan/

DNNを使わず、少量のデータで学習させる独自のアルゴリズムを採用しているそうです。
また、モデルが100%の精度を出すことを期待しているのではなく、人間と役割分担しながら作業効率化を目指しているそうです。

参照：https://ledge.ai/bakery-scan/

ドローンとの連携

ドローンに搭載した画像認識モデルがキャベツを認識し、キャベツの育成状況と収量の予測を実施している事例が紹介されています。
https://ledge.ai/skymatix-cabbage-drone/



画像提供：株式会社スカイマティクス

参照：https://ledge.ai/skymatix-cabbage-drone/

演習

演習1：画像認識技術の活用事例

- 画像認識技術を活用したサービスについて、サービス内容と技術的な仕組みについて調べてみてください。

第8回：回帰問題

アジェンダ

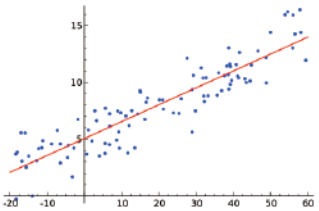
- 回帰
- 正則化
- スパース推定

回帰

回帰とは

[Wikipediaより]
回帰（かいき、英: regression）とは、統計学において、Y が連続値の時にデータに Y = f(X) というモデル（「定量的な関係の構造」）を当てはめる事。別の言い方では、連続尺度の従属変数（目的変数）Y と独立変数（説明変数）X の間にモデルを当てはめること。X が1次元ならば単回帰、X が2次元以上ならば重回帰と言う。Y が離散の場合は分類と言う。

回帰で使われる、最も基本的なモデルは Y = AX + B という形式の線形回帰である。



参照: https://ja.wikipedia.org/wiki/回帰分析

線形回帰とは

- 線形回帰は連続値をとる目的変数yと説明変数x（特徴量）の関係を下記の数式でモデル化します（ $X_0=1$とし、w_0は切片を表します。）。
- 説明変数が一つの場合を単回帰、複数の場合を重回帰といいます。

$$y = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i$$

線形回帰とは

- ボストン市郊外の地域別住宅価格データを使用し、MEDV（住宅価格の中央値）を推定することを考えます。

変数	説明
CRIM	町ごとの一人当たりの犯罪率
ZN	25,000平方フィートを超える敷地に区画された宅地の割合
INDUS	非小売業種の土地面積の割合
CHAS	Charles Riverダミー変数（敷地が川の境界にある場合は1、それ以外の場合は0）
NOX	窒素酸化物の濃度（1000万分の1）
RM	1住戸あたりの平均部屋数
AGE	1940年以前に建設された所有者居住ユニットの割合
DIS	ボストンの5つの雇用センターまでの重み付き距離
RAD	ラジアルハイウェイ（放射状に各方面へ伸びる高速道路）へのアクセスのしやすさの指標
TAX	10,10,000ドルあたりの全額固定資産税率
PTRATIO	町による生徒 - 教師比率
B	1000（Bk - 0.63）^2ここでBkは町による黒人の割合
LSTAT	低所得者の割合
MEDV	住宅価格の中央値（1,000単位）

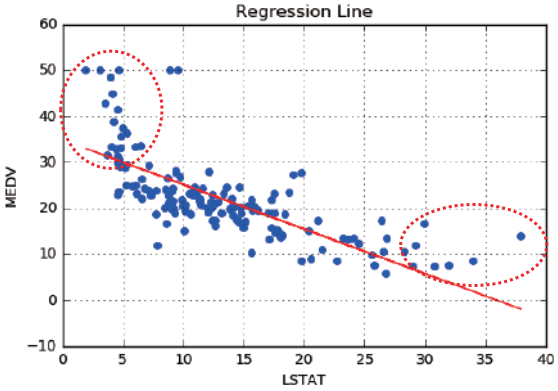
項目間の相関

- 各項目間の相関を計算してみます。
- MEDV（住宅価格の中央値）とRM（1住戸あたりの平均部屋数）は比較的強い正の相関があることがわかります。
- MEDV（住宅価格の中央値）とLSTAT（低所得者の割合）は比較的強い負の相関があることがわかります。

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

単回帰線の確認

- 目的変数である「MEDV: 住宅価格」と比較的強い負の相関がある説明変数である「LSTAT: 低所得者の割合」の散布図と、予測線を図示します。
- LSTATが7～28の範囲では、予測線は住宅価格をよく表現できていますが、その範囲外では乖離があることが確認できます。



単回帰モデルの性能評価

- モデルの性能を評価するために、何かしらの指標を設定したほうが便利です。線形回帰モデルの性能評価として、下記の指標を用いることが一般的です。
  - 平均二乗誤差：残差平方和をデータ数で正規化した値
  - 決定係数：相関係数の二乗

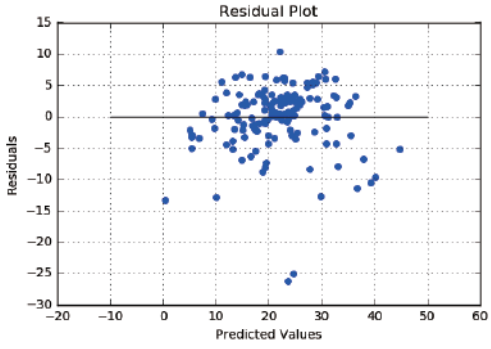
```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

```
r^2 train data:  0.5524780757890007
r^2 test data:  0.5218049526125568
```

決定係数の例。モデル作成に使用した学習データに対する決定係数が、テストデータに対する決定係数より若干高いことが確認できます。

重回帰モデルの精度

- 住宅価格の中央値を全項目から予測する線形回帰モデルを構築すると、決定係数が大幅に改善されていることが確認できます。
- モデルの精度向上のみが目的であれば説明変数を増やして精度向上を図るのは1つの方法ですが、「過学習」の問題が発生することがあります。
- また、モデルの出力結果への説明変数の寄与度を正しく評価できなくなる「多重共線性」の問題が発生することがあります。




```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

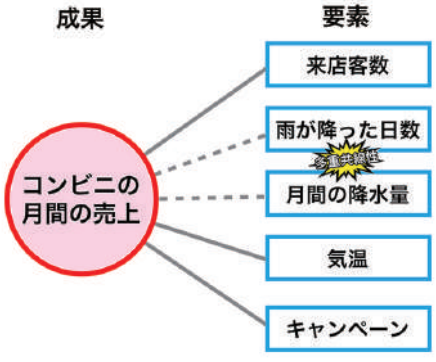
```
r^2 train data:  0.7644563391821222
r^2 test data:  0.6735280865347231
```

多重共線性

- 説明変数を増やしていくと一般的にモデルの表現力が向上し、精度が向上します。
- モデルの精度を高めることのみが目的であれば支障がないこともありますが、モデルの説明性（モデルはなぜそのような予測をしたのか、の説明）が問われる場合、説明変数を闇雲に増やすことには注意が必要です。

多重共線性

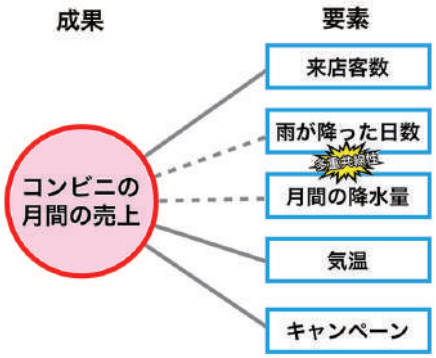
- 説明変数間で相関係数が高い時に多重共線性（multicollinearity）という問題が発生します。
- 多重共線性とは、モデル式の係数が不安定（符号と大きさが安定しない）になり、モデルの予測結果に対する係数の寄与度を正しく評価することができなくなってしまいます。



出展：https://xica.net/vno4ul5p/

多重共線性

- 多重共線性の回避策としては、相関が高い係数のどちらか一方をモデルから外す、ことが一般的です。



出展：https://xica.net/vno4ul5p/

多重共線性の事例

- 説明変数に全項目を使用した重回帰モデルの係数を表1に示します。INDUSとNOXの符号が逆になっているのが確認できます。

表1：重回帰モデルの係数

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.119859	0.0444233	0.0118612	2.51295	-16.271	3.8491	-0.00985472	-1.50003	0.241508	-0.0110672	-1.01898	0.00695273	-0.488111

符号が逆になっている。

表2：全項目の相関係数

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

INDUSとNOXは正の相関がある。

MEDV（住宅価格）に対し、INDUSとNOXは負の相関がある。

過学習、多重共線性の回避

- 過学習や多重共線性を回避するために正則化という手法が存在します。
  - L1正則化：いくつかの説明変数の係数を0にする手法（特徴選択を行っていることになる）です。スパース（疎な）な行列で表現するため、高速に計算できるようになる。
  - L2正則化：各説明変数の係数が大きくなりすぎないようにする（個々の特徴量が出力に与える影響をなるべく小さくした）手法です。

正則化

正則化とは

線形回帰モデルを構築するということは、下式のyとXが与えられたときに、パラメータWを求めていくことです。

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) \quad (\phi_0(\mathbf{x}) = 1)$$

$\phi_j(\mathbf{x})$: 基底関数

目的変数tとの二乗誤差を最小化するパラメータwを求めます。

$$\hat{\mathbf{w}} = \arg\min_{w} \sum_{i} \{t_i - \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_i)\}^2$$

解を求めるためには正規方程式を解くことになります。

$$\hat{\mathbf{w}} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{t} \quad \text{：正規方程式}$$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

参照: https://www.slideshare.net/aratahonda1/prml6-84288779

正則化とは

例えば以下のような行列式を考えます。ここでは意図的に$2X_1 = X_2$としています。

y		Φ			ω		
y1		1	2	4	ω_0		$\omega_0 + 2\omega_1 + 4\omega_2$
y2	=	1	3	6	ω_1	=	$\omega_0 + 3\omega_1 + 6\omega_2$
y3		1	4	8	ω_2		$\omega_0 + 4\omega_1 + 8\omega_2$
		X_0	X_1	X_2			

$$y = \omega_0 X_0 + \omega_1 X_1 + \omega_2 X_2$$
$$= \omega_0 X_0 + (\omega_1 + 2\omega_2) X_1$$

このように特徴ベクトルΦがランク落ちする場合は逆行列が存在しないため、線形回帰の解を求めることができません。

正則化とは

完全に整数倍でない場合でも、ほとんど整数倍に近い場合は逆行列の要素が大きな値になってしまいます。
この特徴ベクトルでωを求めると、極端に大きいもしくは小さな値となってしまいます。

$$\Phi = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 6.1 \\ 1 & 4 & 7.9 \end{bmatrix}$$

$$(\Phi^{T}\Phi)^{-1} = \begin{bmatrix} 6.33 & 18.00 & -10.00 \\ 18.00 & 254.00 & -130.00 \\ -10.00 & -130.00 & 66.67 \end{bmatrix}$$

これが、説明変数間で相関係数が高い時に多重共線性（multicollinearity）が発生してしまう原因です。

正則化とは

回帰は誤差Eを最小にするように係数ωを計算するプロセスです。

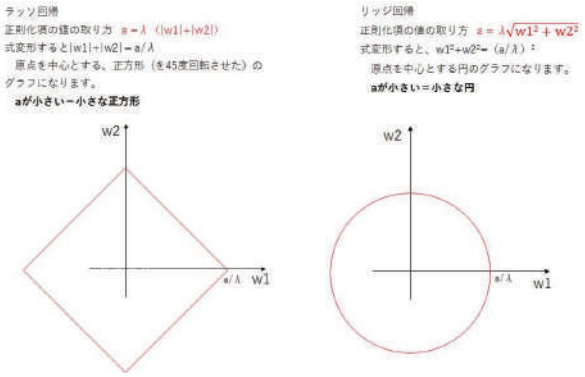
$$E = (y - \Phi(X)\ \omega)2$$

例えば、ωの二乗（正則化項と言います）を加えた下記の式で誤差Eを最小にすることを考えてみます。

$$E = (y - \Phi(X)\ \omega)2 + \alpha\ \omega T \omega$$

ωの値が大きくなることを許してしまうと、Eが大きくなってしまいます。
よって、Eが最小となるようにωを求めると、ωの各成分の絶対値は極端に大きくなることはなくなります。

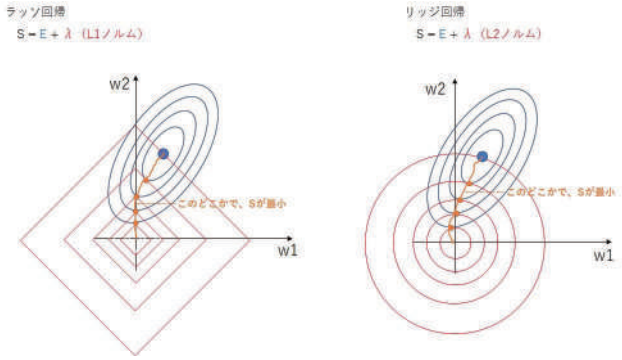
正則化項のとり方によって、ラッソ回帰やリッジ回帰と呼ばれます。



参照：https://qiita.com/MA-fn/items/851f9fc50bda034cd5c6

正則化とは

Eを最小にする過程で前ページの式の α を変えながら計算を繰り返し、第1項が最小になる区間と、第2項（正則化項）が最小になる区間を探索し、両者が一致するところを解とします。



参照：https://qiita.com/MA-fn/items/851f9fc50bda034cd5c6

スパース推定

スパース推定とは

スパース（日本語で「疎」という意味）推定とは、パラメータに0が多くなるように推定することです。

例えば、回帰において回帰係数の推定に使うことができます。
回帰係数が0にならなかった説明変数は、目的変数の推定にとって重要であるとみなすことができます。
このように現象の本質を考察する際に役に立ちます。

また、サンプルサイズよりパラメータが多い場合にはそもそも解析できないという問題（$2X_1 + 3X_2 = 1$のように2変数の方程式は、式が1つでは解が定まらない）も、スパース推定の考え方を使えば解くことができます。
このように、情報量が少なくても解析できるというのもスパース推定の利点です。

スパース推定とは

機械学習で取り扱うスパース性として、大別して下図のように3つのものがあります。

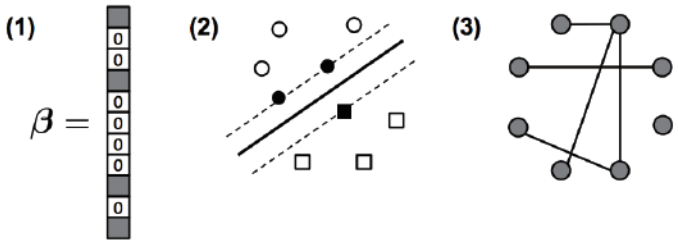


図 1　機械学習で出てくる主な 3 つの種類のスパース性。(1) 少ない変数だけでモデルを表す。(2) 少ない標本だけでモデルを表す。(3) 少ない関係だけに割り切る。

参照：http://ide-research.net/papers/2016_Iwanami_Ide.pdf

少ない変数でモデルを表現する

（1）少ない変数だけでモデルを表す事例について見ていきます。

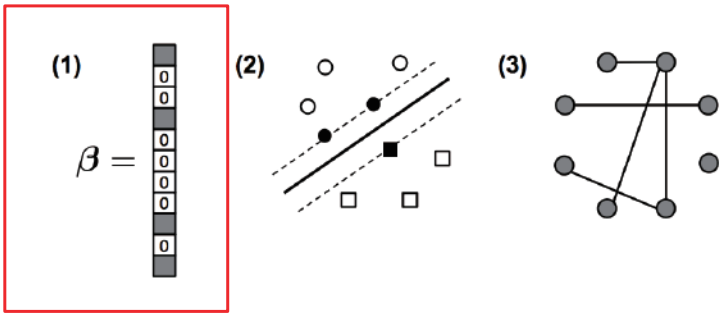


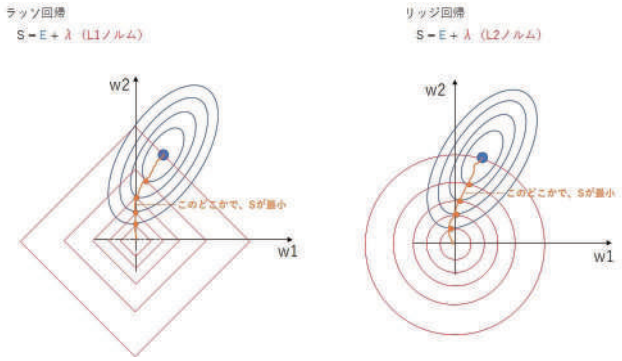
図 1　機械学習で出てくる主な 3 つの種類のスパース性。(1) 少ない変数だけでモデルを表す。(2) 少ない標本だけでモデルを表す。(3) 少ない関係だけに割り切る。

参照：http://ide-research.net/papers/2016_Iwanami_Ide.pdf

L1（ラッソ）正則化

リッジ回帰では最適解でもW1とW2がともに残ってしまいます。
ラッソでは、最適解のところでW1が0になります。

このように重要な項目だけ選別できるのがラッソ回帰です。



参照：https://qiita.com/MA-fn/items/851f9fc50bda034cd5c6

elastic net

lassoは2つの問題があります。

- 相関の高い2つの説明変数が目的変数に関係している場合、一方の説明変数しかモデルに含まれません。本当は重要であるもう片方の変数を捨ててしまうことになります。
- サンプルサイズより説明変数の数の方が大きい場合、サンプルサイズ分の変数までしか選択されません。

この問題を解決するために考案された正則化項がelastic netで、L1ノルムとL2ノルムを合わせた形（α=1の時はラッソ，α=0の時はリッジ）です。
※ノルムとは、ベクトル空間の距離のことです。

$$\frac{1}{2n}\|\boldsymbol{y}-X\boldsymbol{\beta}\|_2^2+\lambda\sum_{j=1}^{p}\left\{\alpha\,|\beta_j|+\frac{(1-\alpha)\beta_j^2}{2}\right\}$$

参照：http://mikutaifuku.hatenablog.com/entry/2018/03/13/231041

少ない標本でモデルを表現する

（2）少ない標本だけでモデルを表す事例について見ていきます。

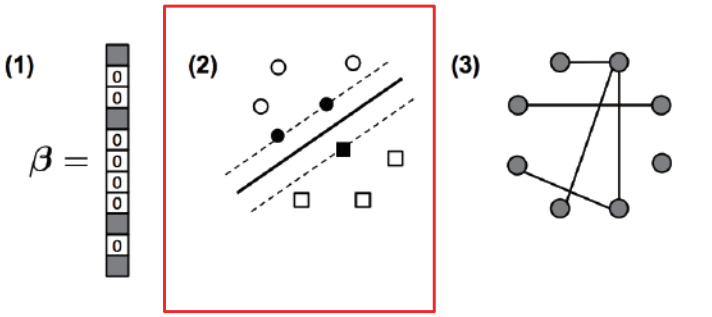
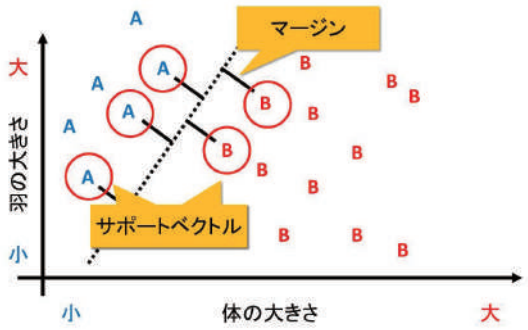


図 1　機械学習で出てくる主な 3 つの種類のスパース性。(1) 少ない変数だけでモデルを表す。(2) 少ない標本だけでモデルを表す。(3) 少ない関係だけに割り切る。

参照：http://ide-research.net/papers/2016_Iwanami_Ide.pdf

SVMによる分類の制約

分類面から遠く、分類面の決定に寄与しない標本の重みはゼロとする考え方があります。

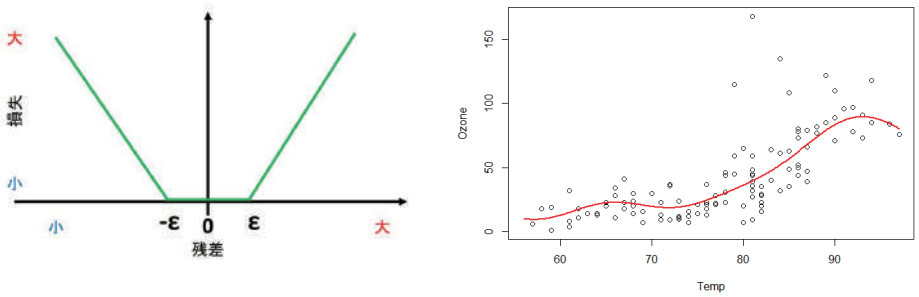


参照：https://logics-of-blue.com/svm-concept/

SVMによる回帰の制約

損失関数を工夫します（ε-不感損失関数）。
残差が−ε～+εまでの範囲内では損失は0、残差がεを超えたら、損失として計上します。
このε-不感損失関数が小さくなるようにモデルを推定します。

この際、残差がεと同じかεを超えるデータをサポートベクトルとみなし、ε-不感損失関数に影響を及ぼさないデータ（サポートベクトルではないデータ）は予測に使われません。



参照：https://logics-of-blue.com/svm-concept/

少ない関係でモデルを表現する

（3）少ない関係だけに割り切る事例について見ていきます。

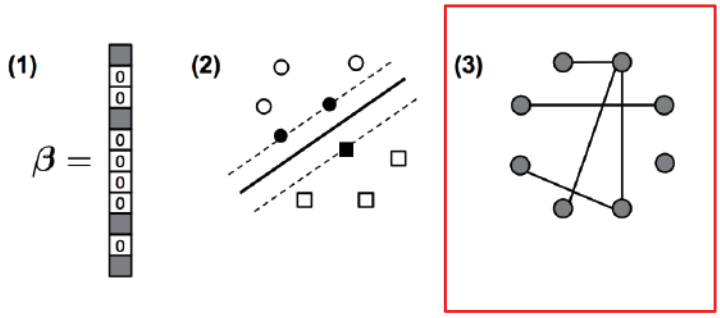
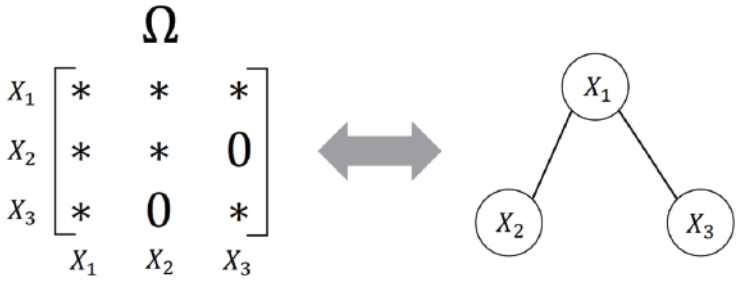


図 1　機械学習で出てくる主な 3 つの種類のスパース性。(1) 少ない変数だけでモデルを表す。(2) 少ない標本だけでモデルを表す。(3) 少ない関係だけに割り切る。

参照：http://ide-research.net/papers/2016_Iwanami_Ide.pdf

ガウシアングラフィカルモデル

ガウシアングラフィカルモデルにL1正則化の考え方を応用したGraphical lassoとよばれる手法があります。
多数の変数間の依存関係を推定するために用いられます。



参照：http://mikutaifuku.hatenablog.com/entry/2018/03/23/230122

HMLasso

欠損値を含むデータから回帰モデルを構築する試みも実施されています。
欠損値を含むデータを機械学習で使用するためには、欠損値を補完するコストが発生します。
また欠損値があっても実施できる回帰手法を実施するためには計算コストがかかります。
このような従来の欠点を補う手法のようです。

[東芝HPより]
収集した製造データに多くの欠損値が含まれている場合でも、品質低下や歩留悪化などの要因を高速・高精度に特定する機械学習アルゴリズム「HMLasso（Least absolute shrinkage and selection operator with High Missing rate）」を開発し、最先端のアルゴリズム「CoCoLasso」と比べ推定誤差を約41％削減することに成功しました。

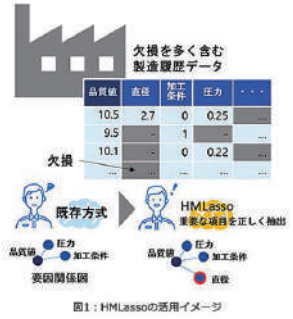


図1：HMLassoの活用イメージ

参照：
https://www.toshiba.co.jp/rdc/detail/1908_01.htm?from=RSS_PRESS&uid=20190802-6221

演習

演習1：回帰

- 単回帰、重回帰とはどのような手法か説明してください。

演習2：多重共線性

- 多重共線性とはどのような現象なのか説明してください。
- 多重共線性の回避策を挙げてください。

演習3：スパース推定

- スパース推定とは何か説明してください。
- スパース推定のメリットを挙げてください。

第9回：時系列分析

アジェンダ

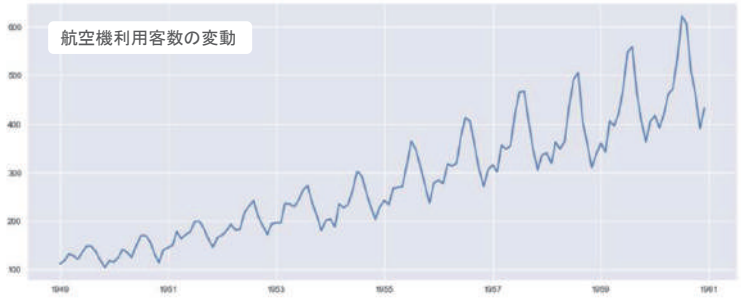
- 時系列データの概要
- 時系列データの概念
- 時系列分析で気をつけること

時系列データの概要

時系列データ

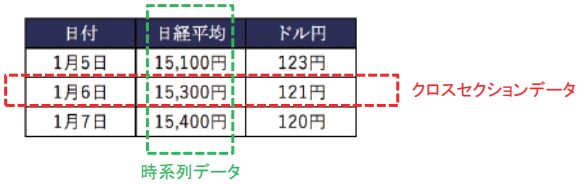
- [Wikipediaより]時系列（じけいれつ、Time Series）とは、ある現象の時間的な変化を、連続的に（または一定間隔をおいて不連続に）観測して得られた値の系列[1]（一連の値）のこと。

- ある1つの項目を時間によって計測したデータのことです。
- 観測される順番に意味があります（時刻tのデータは、時刻t-1などk時間刻み前のデータに何かしら関連している）。



クロスセクションデータ

- ある時点での複数項目のデータをクロスセクションデータといいます。
- 同一時点での複数項目間の分析ができます。

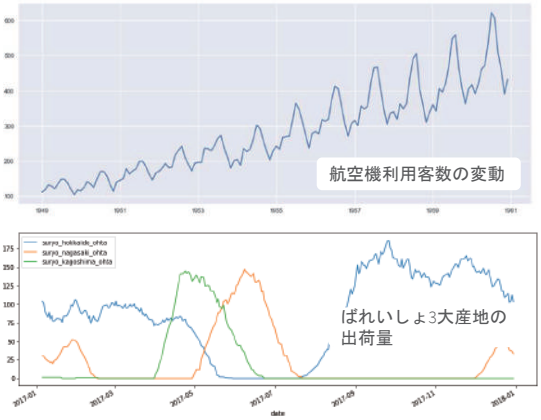


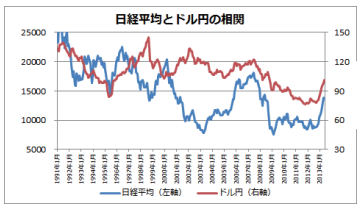
パネルデータ

- クロスセクションデータを時間方向に拡張したデータをパネルデータといいます。
- 単一項目の時系列分析のみでなく、多項目間の連動も加味した分析が可能となります。

時系列データ、パネルデータの事例

- 時間とともに変動する時系列データには、様々な事例があります。

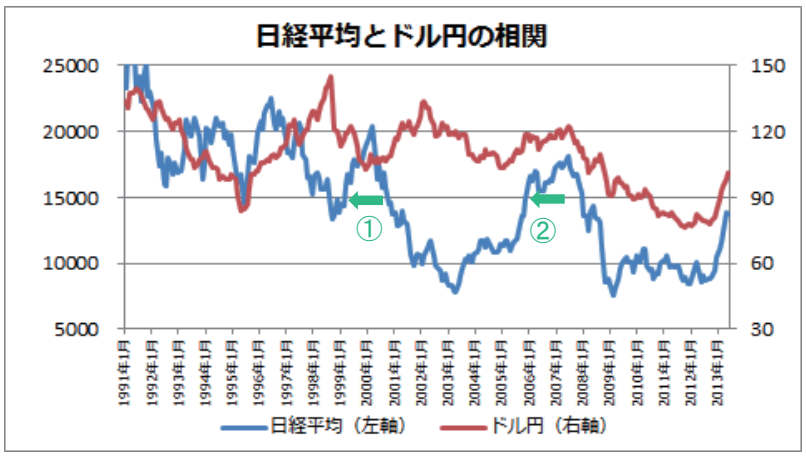




株価と為替レート
の変動

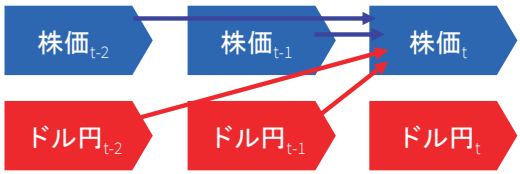
パネルデータの特徴

- ①と②は日経平均が同じ15,000円ですが、反発直後の①と、しばらく上げ続けている②とでは、その後の動きが異なります。
- ①と②でドル円の傾きも異なっています。



パネルデータの分析

- 過去の同じ変数からの影響を考慮する必要があります（自己相関）。
- 違う変数からの影響を考慮する必要があります（多変量）。
- その他、周期性やトレンドも考慮する必要があります。



時系列データの概念

時系列データの種類

原系列
- 時系列データそのもののことを原系列と呼ぶことがあります。

対数系列
- 原系列に対数変換を実施したデータを対数系列といいます。実際のデータには、値が大きくなるにしたがってばらつきが大きくなるものが多数あります。時系列分析では「定常性※」の仮定を満たすようにデータを操作することがありますが、対数変換はよく使われる操作です。

階差系列（差分系列）
- 時間刻みが1つ前のデータとの差分（$y_t - y_{t-1}$）をとったデータを階差系列といいます。実際のデータには「単位根過程※」にしたがうデータが多く、通常はそのままでは扱いません。階差を取ると、単位根過程のデータは「定常過程※」になります。

※各用語は次ページ以降で説明します。

定常過程

[Wikipediaより]
定常過程（ていじょうかてい、英: stationary process）とは、時間や位置によって確率分布が変化しない確率過程※を指す。このため、平均や分散も（もしあれば）時間や位置によって変化しない。
例えば、ホワイトノイズは定常的である。

※確率過程
確率論において、確率過程（かくりつかてい、英語: stochastic process）は、時間とともに変化する確率変数のことである。

定常性

定常性とは同時分布や基本統計量の時間不変性に関する定義です。何を不変とするかにより弱定常性と強定常性に分類されます。

弱定常性

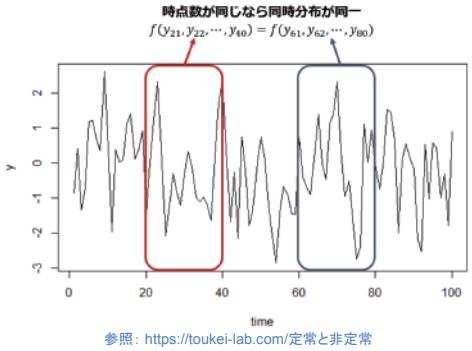
任意の時間tと時間差kについて以下を満たす時、データは弱定常性といいます。

$E(y_t) = \mu$:期待値は一定

$Cov(y_t, y_{t-1}) = E[(y_t - \mu)(y_{t-k} - \mu)] = \gamma_k$:自己共分散は時間差kのみに依存

強定常性

任意の時間tと時間差kについて、$(y_t, y_{t-1},, , y_{t-k})$の同時分布が同一となるとき、データは強定常性といいます。



参照: https://toukei-lab.com/定常と非定常

単位根過程

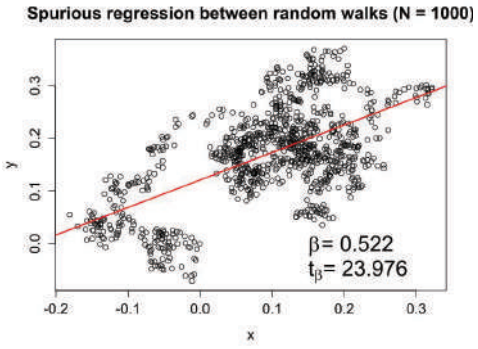
原系列が非定常過程であり、差分系列$\Delta y_t = y_t - y_{t-1}$が定常過程であるとき、その過程は単位根過程といいます。

時系列分析で気をつけること

見せかけの回帰

[Wikipediaより]
見せかけの回帰（みせかけのかいき、英: spurious regression）とは、統計学や計量経済学において、統計的に独立である無関係の二つの時系列変数が最小二乗法による回帰分析において統計的に有意な係数の推定値を取ってしまうという問題である。

右図はランダムウォーク過程の見せかけの回帰を表しています。
シミュレーションで発生させた互いに独立なランダムウォークを
回帰すると、独立なのにも関わらず非常に高いt検定統計量の値と
なります。



参照：https://ja.wikipedia.org/wiki/見せかけの回帰

見せかけの回帰

2つの独立なランダムウォークを考えます。

$$x_t = x_{t-1} + \varepsilon_{1t}, \varepsilon_{1t} \sim iid(0, \sigma_1^2)$$
$$y_t = y_{t-1} + \varepsilon_{2t}, \varepsilon_{2t} \sim iid(0, \sigma_2^2)$$

このとき以下のような回帰モデルを考えます。

$$y_t = \alpha + \beta x_t + \varepsilon_t$$

OLS推定量（最小二乗の推定量）は以下のようになることが知られています。

$$\begin{pmatrix} T^{-1/2}\hat{\alpha} \\ \hat{\beta} \end{pmatrix} \xrightarrow{L} \begin{pmatrix} \sigma_1 h_1 \\ (\sigma_1/\sigma_2) h_2 \end{pmatrix}$$

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} 1 & \int W_2(r)\,dr \\ \int W_2(r)\,dr & \int W_2^2(r)\,dr \end{pmatrix}^{-1} \begin{pmatrix} \int W_1(r)\,dr \\ \int W_2(r) W_1(r)\,dr \end{pmatrix}$$

ここで$\xrightarrow{L}$は分布収束を意味します。またW_1とW_2は互いに独立な標準ブラウン運動です。

見せかけの回帰

前ページの式は$\hat{\alpha}$が$T^{-1/2}$の速度で発散することを示しています。
また、$\hat{\beta}$はある確率変数に収束することを示しています。

x_tとy_tはお互いに独立なので、$\hat{\alpha}$と$\hat{\beta}$は0になるはずです。
ですが、OLS推定量は上のように0には収束しないため$\hat{\alpha}$と$\hat{\beta}$は一致推定量ではないことがわかります。
$\hat{\alpha} = 0$と$\hat{\beta} = 0$でt検定を複数回実施すると、サンプル数Tが大きいときは帰無仮説は殆どの検定で棄却されてしまうことが考えられます（x_tとy_tはお互いに相関があるとみなされてしまいます）。

見せかけの回帰の回避

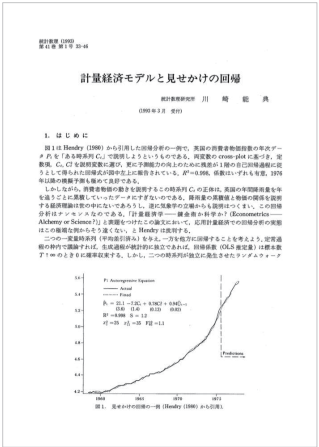
見せかけの回帰を回避する方法は主に2つあります。

1つ目の方法はx_t、y_tそれぞれの差分系列Δx_t、Δy_tを用いて回帰分析することです。

2つ目の方法は回帰式の説明変数にラグ変数（時差のある変数）を追加する方法です。
例えば、y_tに対しx_tだけでなくx_{t-1}を説明変数に追加するということです。

見せかけの回帰についての参考情報

見せかけの回帰について更に詳しく学習したい方は、以下の情報を参照ください。



参照：
https://ismrepo.ism.ac.jp/?action=repository_uri&item_id=31999&file_id=17&file_no=1

共和分

二つの単位根過程 x_t、y_tの線形和$x_t + \beta y_t = z_t$が定常過程に従う時、この二つは共和分の関係を持つといいます。

見せかけの回帰を回避するためには、データの差分を取るという操作をしました。
共和分がある時に差分を取ると、見せかけの回帰とは異なり「本来ならば存在するはずの関係を見落としてしまう」という不都合があります。

共和分

共和分を満たす銘柄の組み合わせを探し出せると、定常過程として取り扱うことができます。
定常過程として取り扱うと時間によって確率分布が変化しなくなるため、ポートフォリオの構築に活かすことができます。
このようなという株の取引戦略を「ペアトレーディング」といいます。

参照: https://www.slideshare.net/ssuserbe0bf0/ss-82038570

演習

演習1：時系列データ

- 時系列データとはどのようなデータか説明してください。

演習2：階差系列

- 階差系列とは、原系列にどのような操作を施したデータか説明してください。

演習3：見せかけの回帰

- 見せかけの回帰とはどのような現象か説明してください。

演習4：共和分

- 共和分とはどのような現象か説明してください。

第10回：アンサンブル学習

アジェンダ

- アンサンブル学習
- アンサンブル学習の応用
- 参考：バイアス・バリアンスと正則化

アンサンブル学習

損失関数

機械学習モデルをトレーニングする際には、損失関数という関数を使用します。
損失関数はモデルの予測値と実際の値の誤差を定義する関数です。
例えば、回帰の場合の二乗誤差は以下の式で定義されます。

$$L(y(x), t) = (y(x) - t)^2$$

ただし、学習データxに対する正解の値がtだとする

損失関数の期待値を展開していくと、以下のようになります。

$$\mathrm{E}\big(L(y(x), t)\big) = \text{バリアンス} + \text{バイアス}^2 + \text{ノイズ}$$

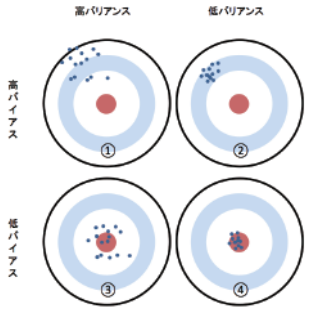
※詳しい展開式に興味がある方は、[https://www.hellocybernetics.tech/entry/2017/01/24/100415]などを参照してください。

バイアスとバリアンス

機械学習器の性能は、バイアスとバリアンスという指標で考えます。

バイアスは実際の値と予測値の誤差の平均のことで、モデルの表現力を表します。
バイアスが大きいと、予測値は真の値から大きくずれてしまいます。
学習データからのトレーニングが十分に進んでいない場合などにバイアスが大きくなります。

バリアンスは、予測値の散らばり具合のことです。
バリアンスが大きいと、予測値は広範囲に分布します。
モデルが学習データに過度に適応してしまった場合（過学習）、
バリアンスが高くなります。



参照：https://www.codexa.net/what-is-ensemble-learning/

アンサンブル学習とは

アンサンブル学習とは、複数の弱学習器（例えば決定木など、簡単に構築できるが精度があまり高くないモデル）を融合させて精度の高い分類や予測を行う手法です。

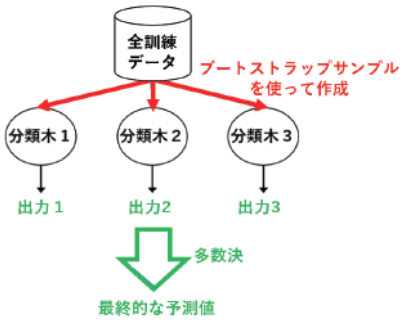
アンサンブルには3つの手法があります。

- バギング
- ブースティング
- スタッキング

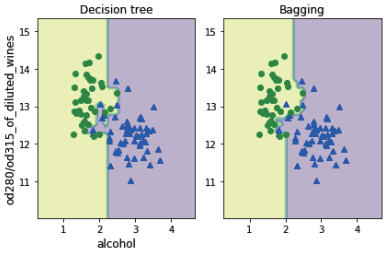
バギング

ブートストラップサンプリングで抽出したデータを使用して独立に多数の弱学習器を作り、それぞれの弱学習器の出力の多数決を取る手法のことです（左図）。

例えば決定木は表現力が高いので過学習しがちですが（バリアンスが高い）、ブートストラップサンプリングすることによって過学習を避ける（バリアンスを下げる）ことができます（右図）。



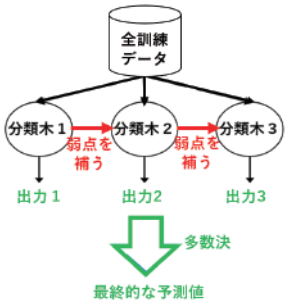
参照：https://mathwords.net/bagging



参照：https://www.investor-daiki.com/it/ai/bagging

ブースティング

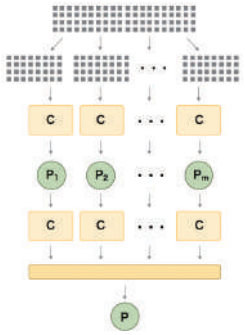
ブースティングはバギングのように弱学習器を独立に作るのではなく、1つずつ順番に弱学習器を作っていきます。順番に弱学習器を作成する際、1つ前の弱学習器の弱点を補うように次の弱学習器を作成するので、バイアスを下げる（モデルの表現力を上げる）効果があります。



参照：https://mathwords.net/bagging

スタッキング

スタッキングは、アンサンブルを複数レイヤーに重ねた構造をしています。
下図の例では、まず第 1 層目に複数の予測モデルからなるアンサンブルを構築します。
2 層目には、1 層目から出力された値を入力とするアンサンブルを構築します。



参照：https://axa.biopapyrus.jp/machine-learning/ensemble-learning.html

アンサンブル学習の応用

バギングの応用：ランダムフォレスト

決定木モデルはバイアスが小さくバリアンスが大きい（過学習を起こしやすい）という性質があります。
ランダムフォレストは、この決定木を複数作りアンサンブル学習させることで、バイアスは小さいままバリアンスを下げることで予測性能を上げる手法です。

バギングと同様にブートストラップ法によって複数の標本集団を作成しますが、通常のバギングとは異なり、それぞれの標本集団に対して、ランダムにx個の特徴量のみを使用して決定木モデルを作成します。

アンサンブル学習では、それぞれの決定木間の相関が高いとバリアンスが下がりづらいことが知られています。ブートストラップ法によって標本を選ぶだけでなく、使われる特徴量も多様になるので、通常のバギングよりも更に多様な標本集団を用いた決定木モデルの作成をすることができ、決定木間の相関が低くなりやすいため通常のバギングよりもバリアンスが低いアンサンブル学習をおこなうことができます。

バギングの応用：ランダムフォレスト

ランダムフォレストのメリット

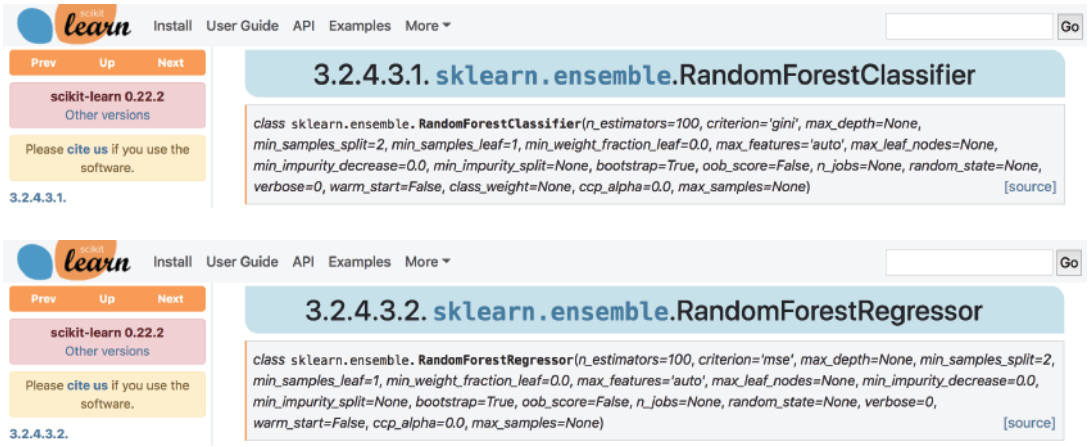
- 決定木がベースとなっているため、特徴量の重要度を計算することができます。
- バギングは並列処理が可能なので 計算が比較的早く済みます。

ランダムフォレストのデメリット

- 特徴量自体が少ない場合は、それぞれの決定木に使用される特徴量をランダムに採択しても似通ってしまうので、バリアンスが下がりづらくなります。
- データ量が少ない場合もブートストラップ法によって復元抽出される標本集団が似通うので相関が高くなりやすく、バリアンスが下がりづらくなります。

バギングの応用：ランダムフォレスト

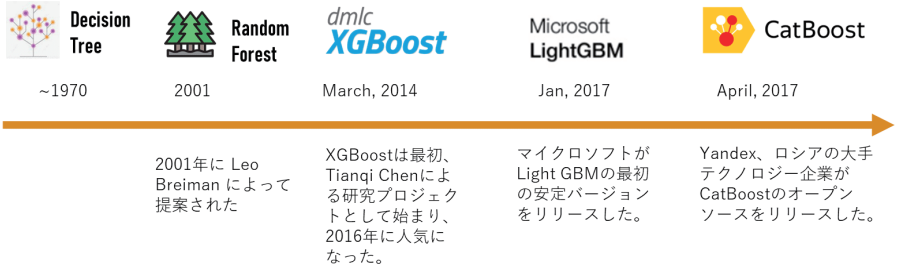
Scikit-learnにはランダムフォレストのクラス分類器、回帰器が搭載されています。
チュートリアルも充実していますので、興味のある方は実際に動かしてみてください。



ブースティングの応用：CatBoost

CatBoostはCategory Boostingの略で、決定木ベースの勾配ブースティングに基づく機械学習ライブラリです。2017にロシアのYandex社からCatBoostが発表されました。
CatBoostは以下の特徴があります。

- 回帰、分類の教師あり学習に対応しています。
- カテゴリカル変数（質的変数）を扱うことができます。
- 過学習を減らして高い精度、学習速度を達成しています。
- GPU、マルチGPUに対応しています。



参照：http://data-analysis-stats.jp/2019/09/29/pythonでcatboostの解説/

ブースティングの応用：CatBoost

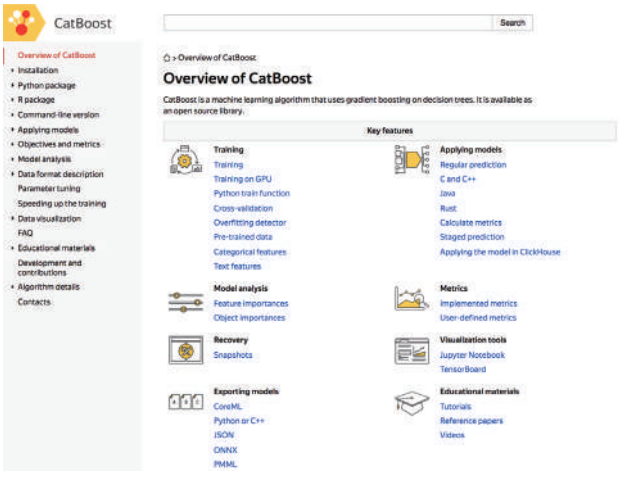
CatBoostとその他のアルゴリズムの性能比較がなされています。
Logarithmic Lossという指標（数値が小さいほど性能がよい）で
XGBoostやLightGBMより性能が高いというレポートがあります。

	CatBoost		LightGBM		XGBoost		H2O	
	Tuned	Default	Tuned	Default	Tuned	Default	Tuned	Default
Adult	**0.26974**	0.27298 +1.21%	0.27602 +2.33%	0.28716 +6.46%	0.27542 +2.11%	0.28009 +3.84%	0.27510 +1.99%	0.27607 +2.35%
Amazon	**0.13772**	0.13811 +0.29%	0.16360 +18.80%	0.16716 +21.38%	0.16327 +18.56%	0.16536 +20.07%	0.16264 +18.10%	0.16950 +23.08%
Click prediction	**0.39090**	0.39112 +0.06%	0.39633 +1.39%	0.39749 +1.69%	0.39624 +1.37%	0.39764 +1.73%	0.39759 +1.72%	0.39785 +1.78%
KDD appetency	0.07151	**0.07138** −0.19%	0.07179 +0.40%	0.07482 +4.63%	0.07176 +0.35%	0.07466 +4.41%	0.07246 +1.33%	0.07355 +2.86%
KDD churn	**0.23129**	0.23193 +0.28%	0.23205 +0.33%	0.23565 +1.89%	0.23312 +0.80%	0.23369 +1.04%	0.23275 +0.64%	0.23287 +0.69%
KDD internet	**0.20875**	0.22021 +5.49%	0.22315 +6.90%	0.23627 +13.19%	0.22532 +7.94%	0.23468 +12.43%	0.22209 +6.40%	0.24023 +15.09%
KDD upselling	**0.16613**	0.16674 +0.37%	0.16682 +0.42%	0.17107 +2.98%	0.16632 +0.12%	0.16873 +1.57%	0.16824 +1.28%	0.16981 +2.22%
KDD 98	**0.19467**	0.19479 +0.07%	0.19576 +0.56%	0.19837 +1.91%	0.19568 +0.52%	0.19795 +1.69%	0.19539 +0.37%	0.19607 +0.72%
Kick prediction	**0.28479**	0.28491 +0.05%	0.29566 +3.82%	0.29877 +4.91%	0.29465 +3.47%	0.29816 +4.70%	0.29481 +3.52%	0.29635 +4.06%

参照：https://catboost.ai/

ブースティングの応用：CatBoost

CatboostはPythonで使用することもできます。
興味のある方は実際に動かしてみてください。



参照：https://catboost.ai/docs/

スタッキングの応用

KDD cup 2015年（データサイエンス国際競技）で優勝したJeong Yoon Lee氏のスタッキング活用事例です。
「64 Single Models」と記載があるステップにおいて、複数の手法でそれぞれデータに対して訓練を行い予測結果を算出しています。
さらに64モデルが出した予測値を入力値とし、「Stage 1 Esenble」と記載があるステップでは新たに15モデルを構築しています。
その後「Stage 2」「Stage 3」モデルをスタッキングしています

Jeong-Yoon Lee, Winning Data Science Competitions

参照：https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/#prettyPhoto

参考：バイアス・バリアンスと正則化

正則化

全ページまででではバイアス・バリアンスを改善しモデルの精度を上げるため、「アンサンブル学習」について学習してきました。
アンサンブル学習以外にも、正則化によってバイアス・バリアンスを改善することができます。

正則化を実施することによってモデルにバイアスが生じますが、バリアンスを改善することができますので、過学習を抑制することができます。
特徴量が増えて複雑なモデルになるほど、正則化が効果を発揮します。

更に詳しく知りたい方は、以下の論文などを参照してください。
[https://www.jstage.jst.go.jp/article/bjsiam/28/2/28_28/_pdf/-char/ja]

正則化とは

線形回帰モデルを構築するということは、下式のyとXが与えられたときに、パラメータWを求めていくことです。

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) \quad (\phi_0(\mathbf{x}) = 1)$$

$\phi_j(\mathbf{x})$: 基底関数

目的変数tとの二乗誤差を最小にするパラメータを求めます。

$$\hat{\mathbf{w}} = \arg\min_{w} \sum_{i} \{t_i - \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_i)\}^2$$

解を求めるためには正規方程式を解くことになります。

$$\hat{\mathbf{w}} = (\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi})^{-1} \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \quad : \text{正規方程式}$$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

参照: https://www.slideshare.net/aratahonda1/prml6-84288779

正則化とは

例えば以下のような行列式を考えます。ここでは意図的に$2X_1 = X_2$としています。

$$\begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 6 \\ 1 & 4 & 8 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \omega_0 + 2\omega_1 + 4\omega_2 \\ \omega_0 + 3\omega_1 + 6\omega_2 \\ \omega_0 + 4\omega_1 + 8\omega_2 \end{bmatrix}$$

$$\begin{matrix} y \\ \\ \end{matrix} = \begin{matrix} \omega_0 X_0 + \omega_1 X_1 + \omega_2 X_2 \\ \omega_0 X_0 + (\omega_1 + 2\omega_2) X_1 \end{matrix}$$

このように特徴ベクトルΦがランク落ちする場合は逆行列が存在しないため、線形回帰の解を求めることができません。

正則化とは

完全に整数倍でない場合でも、ほとんど整数倍に近い場合は逆行列の要素が大きな値になってしまいます。
この特徴ベクトルでωを求めると、極端に大きいもしくは小さな値となってしまいます。

$$\Phi = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 6.1 \\ 1 & 4 & 7.9 \end{bmatrix}$$

$$(\Phi^T\Phi)^{-1} = \begin{bmatrix} 6.33 & 18.00 & -10.00 \\ 18.00 & 254.00 & -130.00 \\ -10.00 & -130.00 & 66.67 \end{bmatrix}$$

これが、説明変数間で相関係数が高い時に多重共線性（multicollinearity）が発生してしまう原因です。

正則化とは

回帰は誤差Eを最小にするように係数ωを計算するプロセスです。

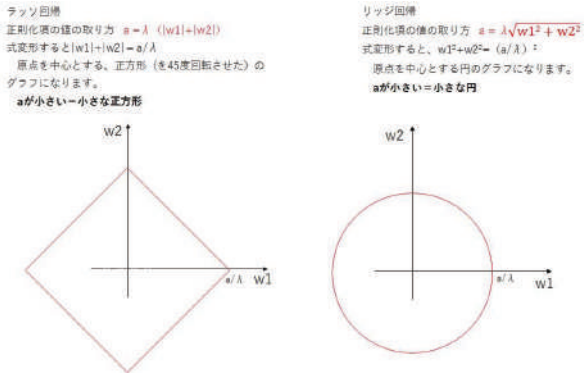
$$E = (y - \Phi(X)\ \omega)2$$

例えば、ωの二乗（正則化項と言います）を加えた下記の式で誤差Eを最小にすることを考えてみます。

$$E = (y - \Phi(X)\ \omega)2 + \alpha\ \omega T \omega$$

ωの値が大きくなることを許してしまうと、Eが大きくなってしまいます。
よって、Eが最小となるようにωを求めると、ωの各成分の絶対値は極端に大きくなることはなくなります。

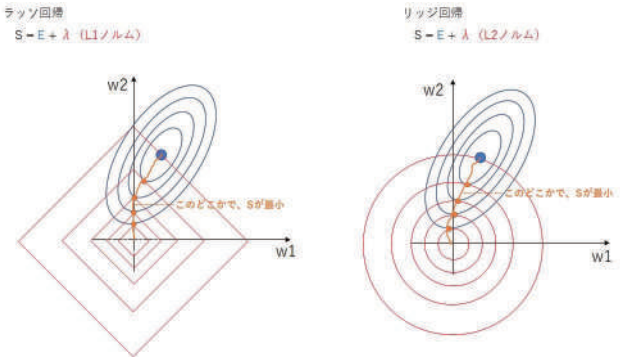
正則化項のとり方によって、ラッソ回帰やリッジ回帰と呼ばれます。



参照：https://qiita.com/MA-fn/items/851f9fc50bda034cd5c6

正則化とは

Eを最小にする過程で前ページの式のαを変えながら計算を繰り返し、第1項が最小になる区間と、第2項（正則化項）が最小になる区間を探索し、両者が一致するところを解とします。



参照：https://qiita.com/MA-fn/items/851f9fc50bda034cd5c6

演習

演習1：バイアスとバリアンス

- モデルの評価に用いられるバイアスとバリアンスとは何か、説明してください。

演習2：バギング

- アンサンブル学習の1つであるバギングについて、どのような手法か説明してください。

演習3：ブースティング

- アンサンブル学習の1つであるブースティングについて、どのような手法か説明してください。

演習4：スタッキング

- アンサンブル学習の1つであるスタッキングについて、どのような手法か説明してください。

第11回：ニーズ予測

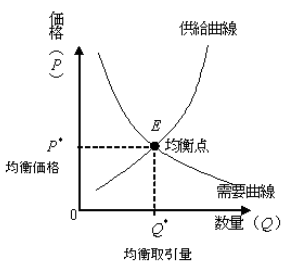
アジェンダ

- ニーズ予測（需要予測）とは
- 需要予測の例（客室予約数の予測）
- 説明変数の重要度を測る

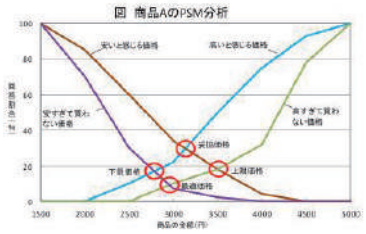
需要予測とは

需要予測とは

- 需要予測とは、モノやサービスの需要を短期的または長期的に予測することです。
- 需要の変動は供給量/価格の変動と密接な関係があります。



引用：需給曲線
https://ja.wikipedia.org/wiki/需要と供給



引用：価格感応度
https://www.stat.go.jp/koukou/trivia/careers/career5.html

需要の変動要因

すべてのモノ、サービスに共通
- 供給量、価格

食品（主食、副菜などの消費変動が小さいもの）
- 産地の気象条件、人口動態

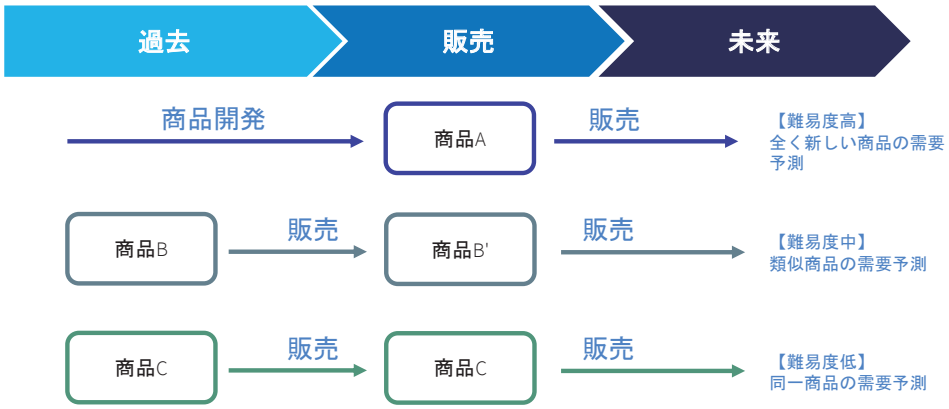
食品（お菓子など消費変動が大きいもの）
- 消費地の気象条件、立地、トレンド

消費財（家庭で消費されるいろいろ）
- 家族構成、生活パターンなど様々な要因が影響する。

その他（ホテルの部屋、イベントチケットなど）

需要予測の難易度

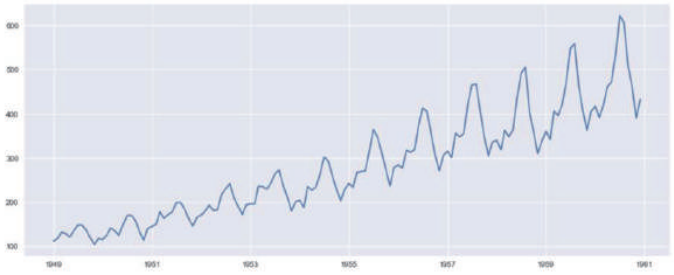
過去に同じ商品の販売履歴がある場合、データが蓄積されているので需要予測は比較的容易です。
新商品の販売など、過去のデータが参考にならない場合は、需要予測は難しくなります。



需要予測の例
（難易度低：同一商品の需要予測）

同一商品の過去の売上データなどを元に、単変量自己回帰を実施します。

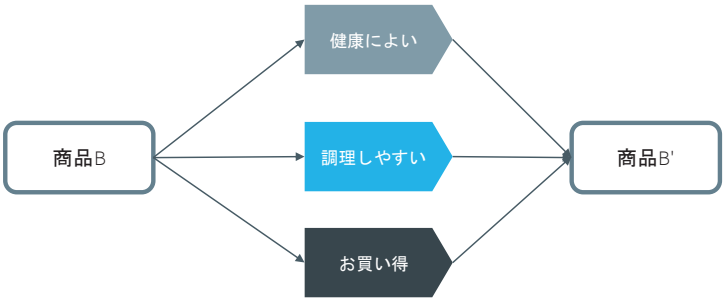
例えば、「ユニクロ横浜みなとみらい店におけるフリースの売上データの過去3年分を元にモデルを作成する」などです。自己回帰モデルを使用することで季節性を考慮した予測ができます。



引用：時系列データのイメージ
https://logics-of-blue.com/python-time-series-analysis/

需要予測の例
（難易度中：類似商品の需要予測）

過去の売上実績がある類似商品を商品DNA（商品に付与した特徴を人の遺伝子であるDNAに見立てた言葉）に分解し、商品DNAを元に今後売り出す商品の需要を予測します。



需要予測の考え方
（難易度高：全く新しい商品の需要予測）

全く新しい商品の需要予測は非常に高度な作業です。需要予測サービスを提供する会社ごとにさまざまなノウハウがあり、一概にどの手法が優れているとは言えません。
（参考）https://www.intage.co.jp/gallery/data-science3/
（参考）https://www.cross-m.co.jp/solution/conceptdevelopment/

一般的に、需要予測モデルは一度構築したら終わりではなく、予測の予実差を元に予測精度を向上し続けるための更新作業を継続します。

需要予測の例（客室予約数の予測）

レベニューマネジメントとは

ホテルの客室や航空機の座席を例に考えます。
レベニューマネジメントとは、「需要予測を基に客室（座席）販売数と価格を調整し、ホテル（航空機）全体で収益の最大化を目指す手法」と言えます。

ホテルや航空機では、今日売れ残る客室や座席を、明日販売する事は出来ません。
なるべく安く販売して、売れ残りを少なくするというのも一つの考え方です。
ですが、もっと高い価格でも買ってくれたかもしれない客（どうしても、その日に、その場所の周辺で宿泊したかったビジネス客など）を逃してしまうことになり、機会損失が発生します。

収益の最大化とは、このように需要を基に商品の価格と数を調整する必要があります。

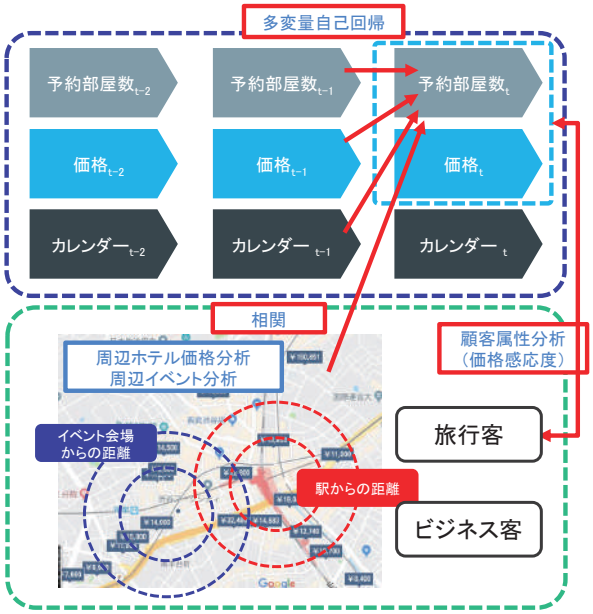
需要（予約部屋数）の推定

簡単なレベニューマネジメントモデルを考えてみます。

例えば、今日の予約部屋数が多ければ、明日には部屋が埋まってしまうかもしれないので、価格を上げるのは自然なことに思えます。

また、近くでイベントがあったり、夏休みなどの休日には需要が増えることが予測できるので、価格が上がるかもしれません。

一人で宿泊する客はビジネス客が多いと思われるので、価格にはあまり敏感ではないかもしれません（会社の経費で精算するので、自分のお金ではない）。

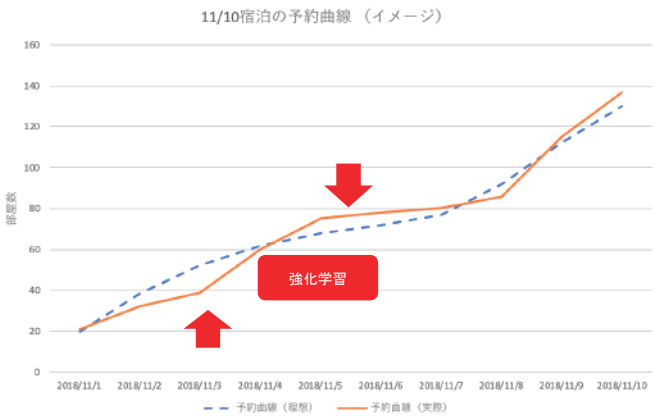


理想予約曲線の作成と強化学習

前ページで記載したような変数を使用し、需要予測モデルを作成したとします。
- 説明変数：月・曜日・宿泊日までの日数、部屋単価
- 目的変数：単価当たり予約部屋数

実際の運用の中で価格を上げ下げして、実際の予約数がどのように変動するのか実験します。

理想的な宿泊の予約曲線と、実際の予約曲線の乖離を解消する強化学習を実行し続けます。



その他の説明変数

需要の予測をするレベニューマネジメントモデルに使用する説明変数として、部屋の価格や予約数、カレンダーなどを挙げてきました。
最近のレベニューマネジメントモデルには、
- 口コミ
- ライバルホテルの予約数
- イベント情報

などを駆使しているモデルも多数あるようです。

どのようなデータが利用されているか

レベニューマネジメントを販売している各社の比較です。
汎用モデルとしてパッケージングされているものから、ホテル独自にチューニングする特化型のものまで、広く存在しています。
また、使用する変数や価格設定も各社ともに工夫しているようです。

	A社	B社	D社	E社
競合データ収集（ホテル）	○		○	○
競合データ収集（民泊）	○		―	―
イベントデータ収集	○		○	○
ホテル建設情報	○		―	―
可視化	○		○	○
価格レコメンド	○		○	○
料金	初期費用あり 月額料金：一部屋ごと。部屋数に上限なし。	要件次第	初期費用あり 月額料金：100室以下、101室以上で部屋ごと料金が異なる。	初期費用あり 200室以下、201室以上で部屋ごと料金が異なる。

レベニューマネジメントモデルの応用分野

これまではホテルの宿泊予約数を例としてレベニューマネジメントモデルの話を記載してきました。
「今日売れ残る客室や座席を、明日販売する事はできない」という特性の製品が存在する分野では、同じ考え方を適用できる可能性があります。

例えば、以下の表のような分野でレベニューマネジメントモデルを導入して、収益を改善できる可能性があります。

	レンタカー	高速バス	新幹線	飛行機	有料駐車場	貸し会議室	ゴルフ場	ホテル
在庫の単位	車	座席	座席	座席	スペース	部屋	時間枠	部屋
在庫の動かしやすさ	容易	困難	困難	困難	困難	困難	困難	困難
キャパシティ	変動	固定	固定	固定	固定	固定	固定	固定
変動費割合	？	？	？	？	？	？	？	？

説明変数の重要度を測る

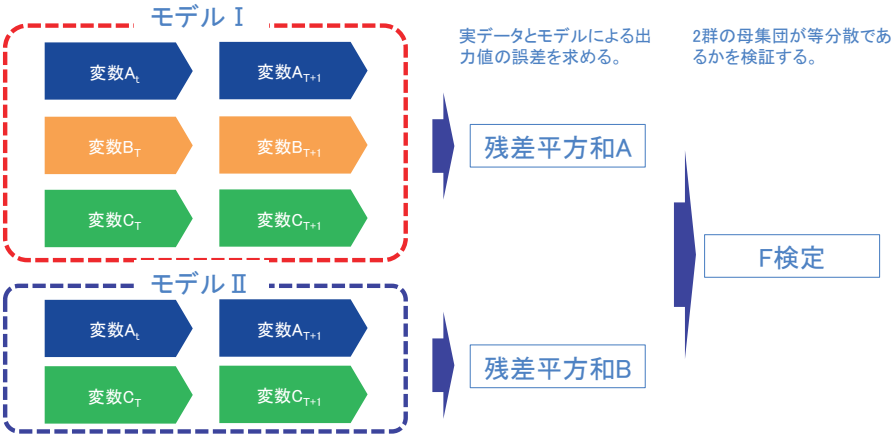
どの変数の影響が大きいのか？

機械学習モデルを実ビジネスで活用する上で避けて通れないのが、モデルの説明性を高めるというプロセスです。
ビジネスの現場では、リソース（人・モノ・金）を投入するための判断が求められます。
いくら精度の良いモデルが存在したとしても、「モデルはなぜそのような判断をしたのか」という説明ができなければ、判断は下しにくいというのが実情です。

線形モデル、非線形モデル、時系列モデルなど多種多様なモデルにおいて、説明変数の重要度を測るための様々な手法が存在します。

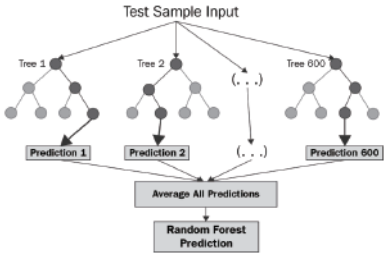
変数の重要度を見積もる： 時系列モデル

時系列データにおける因果の分析に使われるのがGranger causality（グレンジャー因果性）検定です。
モデルⅠではA/B/Cという3つの変数を使用します。
モデルⅡでは、Bを落としてA/Cという2つの変数を使用します。
モデルⅠの精度がモデルⅡに比べて大幅に良ければ、変数Bは重要であるとみなします。

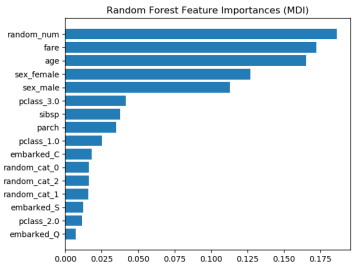


変数の重要度を見積もる：ランダムフォレスト

ランダムフォレストのような決定木ベースのアンサンブル分析器では，特徴量の重要度を算出することができます。
ある特徴量について、データの並び順をぐちゃぐちゃにし、ぐちゃぐちゃにする前と後で、決定木の精度が変わるかどうか比較します。
精度が大きく変わったら重要な特徴量、変わらなかったら重要でない特徴量とみなします。



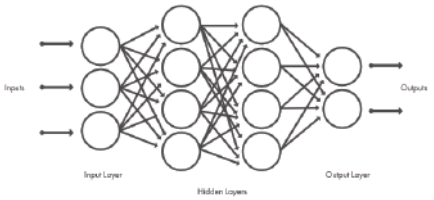
引用：https://www.oreilly.com/library/view/tensorflow-machine-learning/9781789132212/d3d388ea-3e0b-4095-b01e-a0fe8cb3e575.xhtml



引用：https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html

変数の重要度を見積もる：ディープラーニング

LIMEはディープラーニングなどの複雑なモデルを、単純な線形回帰で近似することで解釈性の向上を目指すという手法です。
ですが複雑な非線形のモデルを線形回帰で近似するというのは限界がありますので、重要度を見積もりたい対象データの周辺のデータをサンプリングし、そのデータを教師データとして局所的な線形回帰モデルを作成します。



引用：ディープラーニング
これだけは知っておきたい3つのこと
https://jp.mathworks.com/discovery/deep-learning.html

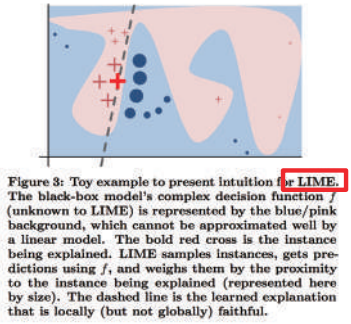


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

引用："Why Should I Trust You?"
Explaining the Predictions of Any Classifier
https://arxiv.org/pdf/1602.04938.pdf

演習

演習1：需要予測

- 需要予測とは何か、説明してください。
- 需要予測が応用されている事例を探し、目的変数と説明変数がそれぞれなにか整理してください。

演習2：需要予測の難易度

- 需要予測はどのようなときに難しくなるか説明してください。
- 需要予想の精度を上げるために、どのようなアプローチが考えられるか、議論してください。

演習3：特徴加工

- ホテルの宿泊予約数を予測するために、レベニューマネジメントモデルを使用しているとします。更に精度を上げるため、口コミデータも使用することにしました。
- 機械学習モデルで使用できるようにするため、口コミデータをどのように加工すればよいでしょうか？ベクトル化する、キーワードを抽出するなど、答えは無数にありますので、自由な発想で考えてみてください。

演習4：変数の重要度

- 本資料で扱った手法以外にも、変数の重要度を測る手法が多数存在します。他にどのような手法があるのか、調査してください。

第12回：異常検知

アジェンダ

- 異常検知の考え方
- 異常検知の事例

異常検知の考え方

異常検知とは

[Wikipediaより]
異常検知（いじょうけんち、英: anomaly detection）や外れ値検知（はずれちけんち、英: outlier detection）とは、データマイニングにおいて、期待されるパターンまたはデータセット中の他のアイテムと一致しないアイテムやイベントや観測結果を識別すること。
何が異常であるかを定義するのは、タスク次第ではあるものの、Varun Chandolaら[1]は異常というのは通常の動作として明確に定義された概念に準拠しないデータパターンである定義している。
各タスクに適用すると通常、異常とはは銀行詐欺（英語版）、クレジットカード不正利用、構造欠陥、医学的な問題、文書中の誤り検出、不審な行動検出、機械の故障検知などの問題に翻訳する。なお、異常（anomaly）は、外れ値（outlier）、珍しい物（novelty）、雑音（noise）、変動（deviation）、例外（exception）などとも呼ばれる。

異常検知の基本的な考え方

例えばクレジットカードの不正検知を考えてみます。
不正利用により発生するデータは、通常利用により発生するデータと比較して圧倒的に少なくなります。
このようにデータが偏っている場合、教師あり学習で異常検知モデルを作るのは難易度が高くなってしまいます。

データの偏りを補正せずにモデルを構築すれば、ほとんどすべてのデータに対し「正常利用」と判定してしまうでしょう。
正常/異常のデータ件数を補正してモデルを構築すれば精度はいくらかは向上するかもしれませんが、正常データの件数を大幅に絞るため、正常データの分布のすべてをカバーしたモデルを構築することが難しくなってしまいます。

異常検知の基本的な考え方

前述のような教師あり学習が難しい場合、「正常なデータ」ではないデータを異常とみなして検知する方法が取られます。

「正常なデータ」とはなにか、を決めるのによく使われるのが確率分布です。
正常なデータのサンプルから確率分布を推定することができれば、すべての正常データを手に入れることができなくても、正常データが取りうる範囲を予測できるようになります。

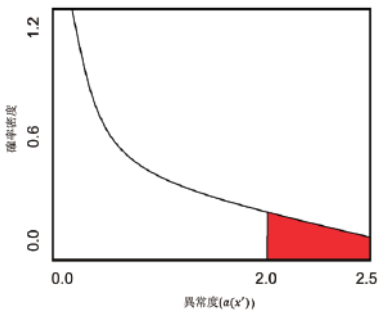
正常データの分布から外れたデータを、異常データとみなします。

※この手法の欠点は、正常データの異常データが紛れ込んだ状態で確率分布を推定してしまうと、正常データの確率分布が正しく推定できなくなってしまうことです。

ホテリング理論

- ホテリング理論では異常度を右のような式で定義します。σは標準偏差、μは平均を表しています。
- ホテリング理論ではデータが正規分布に従っている事を仮定しています。異常度はデータ数が十分に大きければ自由度1のカイ二乗分布に従うとしています。
- 右図に示されたカイ二乗分布では曲線で囲まれた面積が確率を表しています。面積が小さければごく稀に発生する異常時データである可能性が高いと言えます。

$$a(x') = \left(\frac{x' - \hat{\mu}}{\hat{\sigma}} \right)^2$$



出展：
https://qiita.com/Zepprix/items/f6a5de2e3f6689bd2c1f

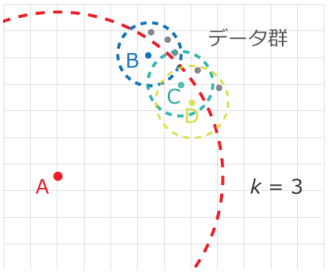
ホテリング理論の問題点

- データが単一の正規分布から発生していると仮定しています。正規分布から著しく外れているデータの場合や分布が複数の山を持つ場合などは、異常値を正しく判断できなくなります。
- 正規分布のパラメータは変化しないと仮定しているため、分布のパラメータが変化する時系列データのようなデータには適用することができません。

局所外れ値因子法（LOF法）

- 右式の局所密度を利用して異常検知をするのがLOF（Local Outlier Factor ）法です。
- 外れ値である点Aを基準とすると、A自身の局所密度は低く、近傍点である点B, C, D の局所密度は高くなっています。自身の局所密度と近傍点の局所密度が等しいときほど正常データであり、その差が大きいほど外れ値である可能性が高いと解釈できます。

$$\text{局所密度} = \frac{1}{\text{近傍}k\text{個の点との距離の平均}}$$



出展：
https://qiita.com/Zepprix/items/f6a5de2e3f6689bd2c1f

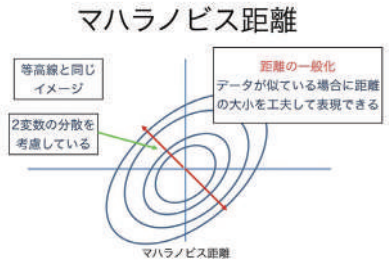
マハラノビス距離

マハラノビス距離とは統計学で用いられる距離の一種で、普通の距離（ユークリッド距離）を一般化したものです。マハラノビス距離は以下のように表されます。

$$\sqrt{(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)}$$

x_i：i番目のデータ、μ：データ列の平均値ベクトル、Σ^{-1}：分散共分散行列

マハラノビス距離は上の式からもわかるように、ユークリッド距離とは異なりデータ間の相関を考慮した式となっています。



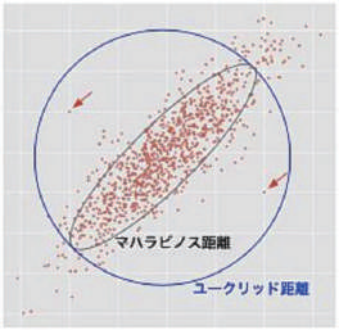
出展：
https://qiita.com/shopetan/items/ceb7744facc21c3881d2

マハラノビス距離

マハラノビス距離に何らかの閾値θを設定し、閾値θを超えた遠い距離にあるデータは異常値とみなします。

$$\theta < \sqrt{(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)}$$

マハラノビス距離は平均と分散という概念を定式化しています。
平均値は異常値の影響を大きく受けてしまいますので、正常データだと思っていたデータセットに異常値が含まれてしまうと、うまく異常検知ができなくなってしまいます。
このような問題をクリアするために、中央値を利用する方法もあるようです。



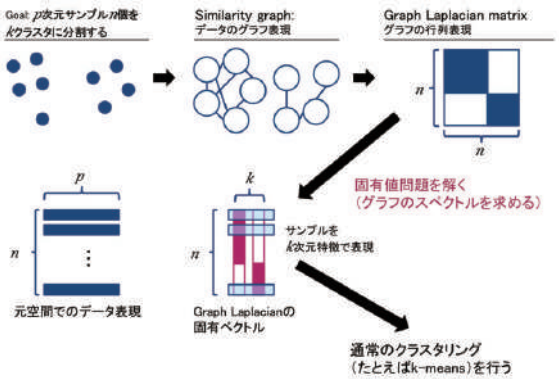
出展：
https://monoist.atmarkit.co.jp/mn/articles/1912/04/news020.html#l_sp_191204mltips03_01.jpg&_ga=2.233810654.1186856895.1586481114-2107126371.1583624282

スペクトラルクラスタリング

スペクトラルクラスタリングはクラスタリングアルゴリズムの一つです。
データからグラフを生成し、グラフの連結成分分解を応用してクラスタリングします。

※スペクトルクラスタリングの理論に興味のある方は、以下のURLなどを参照してください。

- スペクトラルクラスタリング入門[https://techblog.nhn-techorus.com/archives/5464]
- グラフラプラシアンを噛み砕いて噛み砕いて跡形もなくしてみた[https://qiita.com/silva0215/items/0d1d25ef51b6865a6e15]
- スペクトラルクラスタリングの話[https://mr-r-i-c-e.hatenadiary.org/entry/20121214/1355499195]

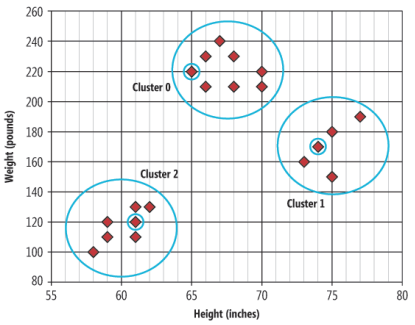


出展： https://mr-r-i-c-e.hatenadiary.org/entry/20121214/1355499195

スペクトラルクラスタリング

クラスタリングした各クラスターにから遠い距離にあるデータを異常データとみなします。

例えば、クラスターごとに重心を求め、それらの重心から一定の距離を超えた場所にあるデータを異常とみなす、というように判定します。

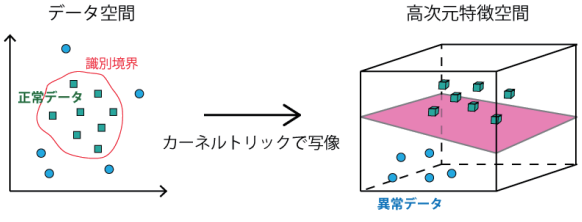


出展：
https://docs.microsoft.com/ja-jp/archive/msdn-magazine/2013/february/data-clustering-detecting-abnormal-data-using-k-means-clustering

One Class SVM

クラス分類などで使用されるサポートベクターマシンを異常検知に応用する方法もあります。
正常データのみを学習させるのですが、その際に正常データをクラス0、原点のみをクラス1というようにラベリングします。

カーネルトリックと呼ばれる手法を用いて高次元空間の特徴空間へデータを写像すると、正常データは原点から遠くに配置されるように写像され、異常データは原点の近くに集まるようになります。



出展：
http://hktech.hatenablog.com/entry/2018/10/11/235312

オートエンコーダー

オートエンコーダはデータを入力層で次元圧縮し、出力層で復元します。
次元圧縮した際に重要な特徴のみを抽出することができれば、うまく復元することができます。
この圧縮・復元を正常データのみを教師データとして学習を繰り返すことにより、オートエンコーダーは正常データをよく表す特徴を学習します。

学習済のオートエンコーダは教師データによく似たデータ(正常データ)を入力した場合、復元が上手くいきます。
一方，教師データと大きく異なるデータ(異常データ)を入力した場合、うまく復元されません。

復元した結果がどれくらい正常データと似通っているか、という基準で異常データを検出します。

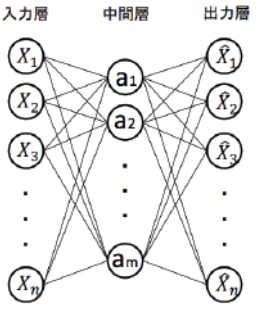


図２　オートエンコーダ概略図

出展：
https://www.gitc.pref.nagano.lg.jp/reports/pdf/H29/H29P41.pdf

オートエンコーダー

VAE(Variational AutoEocoder：変分オートエンコード)はオートエンコーダーを確率的に扱えるようにしたものです。このVAEを活用した異常検知も行われています。

興味のある方は以下の論文を参照してください。
深層学習による胸部X線写真からの診断補助
[https://www.ai-gakkai.or.jp/jsai2017/webprogram/2017/pdf/773.pdf]

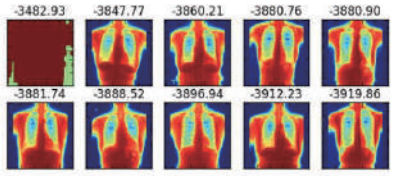


図 6: 診断異常画像での VAE 変分下界 (ベスト 10)

出展：
https://www.ai-gakkai.or.jp/jsai2017/webprogram/2017/pdf/773.pdf

異常検知の事例

事例：オートエンコーダを用いた工具摩耗の検知

旋盤における工具の切削力をデータとして、工具の摩耗具合を検知するという取り組みです。
従来の光学的な検知方法と比較した議論が記載されています。

オートエンコーダを使うことにより微小な摩耗が検知できたと報告されています。

オートエンコーダを用いた工具摩耗の検知

新村 諭*1　新井 亮一*1

Detection of Tool Wear with the Autoencoder

Satoru SHINMURA and Ryoichi ARAI

NC旋盤やマシニングセンタ等のNC工作機械の普及により，国内の切削加工現場は，ほぼ自動化されている。切削の自動運転中に，工具摩耗により工具が折損した場合，そのまま加工を継続すると，工作物の損傷や工作機械の故障につながる危険性がある。そこで，本研究ではニューラルネットワークの一種であるオートエンコーダで工具摩耗を検知し，これら無人運転中の危険性を低くすることを試みた。NC旋盤による実験の結果，切削回数の増加に伴い，工具の異常度が上昇する傾向が見られた。また，突発的な異常も検知することができた。

キーワード：オートエンコーダ，工具摩耗，ニューラルネットワーク

出展：https://www.gitc.pref.nagano.lg.jp/reports/pdf/H29/H29P41.pdf

事例：深層学習による胸部X線写真からの診断補助

肺のX 線写真を中心とするデータから肺の異常検知を行う方法について議論されてます。
CNNとVAEを使った手法について記載されています。

深層学習による胸部X線写真からの診断補助

Diagnosis support from Chest X-ray pictures with Deep Network

黒滝 紘生 *1 Hiroki Kurotaki　中山 浩太郎 *1 Kotaro Nakayama　上原 雅俊 *2 Masatoshi Uehara　山口 亮平 *3 Ryohei Yamaguchi　河添 悦昌 *4 Yoshimasa Kawazoe
大江 和彦 *3 Kazuhiko Ohe　松尾 豊 *1 Yutaka Matsuo

*1 東京大学工学系研究科技術経営戦略学専攻
The Department of Technology Management for Innovation, Graduate School of Engineering, The University of Tokyo
*2 東京大学工学部計数工学科
Department of Mathematical Engineering and Information Physics, School of Engineering, The University of Tokyo
*3 東京大学大学院医学系研究科 Graduate School of Medicine, The University of Tokyo
*4 東京大学医学部附属病院 The University of Tokyo Hospital

X-ray pictures of the chest are used to detect abnormalities or diseases (e.g. rib fracture, lung cancer, pneumonia etc). It is beneficial if we managed to support clinical judgement by doctor automatically with machine learning models. We propose a method to detect abnormal images from chest X-ray images for both unsupervised and supervised settings. For unsupervised task we used variational autoencoder (VAE) and for supervised we used convolutional network (CNN). We verified our method with a chest X-ray image dataset provided from The University of Tokyo Hospital. Our method successfully discriminated the abnormal images from the normals with high accuracy.

出展：https://www.ai-gakkai.or.jp/jsai2017/webprogram/2017/pdf/773.pdf

事例：ワークスタイルの変調を検知

勤怠システム上で取得可能な13の勤怠項目に対して、モデルが定常的な勤怠行動を学習していると記載があります。
その学習結果に対して、直近の勤怠行動に変調的な行動が発生した時、労務担当者にアラートをあげる仕組みとしているそうです。

HuRAid 勤怠分析 勤怠管理 会社概要 お知らせ 採用情報 お問い合わせ

異常検知アルゴリズムを利用した「ワークスタイルの変調」を検知する新機能の提供を開始

従業員1人1人の働き方の変化を把握し、多様で柔軟な働き方の定常化をサポート

2019.10.08

HuRAid株式会社（フレイド株式会社　本社：東京都台東区 代表取締役 鈴木原彦 以下、HuRAid）は、働き方改革をサポートするリテンションクラウドサービス「HuRAid勤怠分析」の新機能として、「ワークスタイルの変調」を検知する機能の提供を開始いたしました。

「ワークスタイルの変調」検知機能開発の経緯

これまでHuRAid勤怠分析では、「従業員ごとの離職リスクの視覚化機能」・「勤怠分析機能」・「ストレスチェック機能」等を提供し、導入企業様においては離職防止・メンタル不調防止・勤怠マネジメント強化をサポートしてまいりました。

一方で、昨今の働き方改革における従業員1人1人の事情に応じた多様で柔軟な働き方に対して、上記の課題・機能もより従業員個人の働き方に即した分析サービスの提供の必要性も強く感じておりました。例えば、働き方改革の重要事項である長時間労働の抑止は、メンタル不調・離職率の低下、従業員満足度の向上に繋がる側面があるが、その一方で「もっと働きたい・会社に貢献したい」という従業員のモチベーションの阻害要因にも繋がるのではないかと考えています。事実、当社が提供しているサービス内で離職リスクの高い従業員の残業時間は中期的な減少傾向があることが分かっております。重要なのは、従業員1人1人のワークスタイルを適切に常態化することにあるのではないかと考えております。

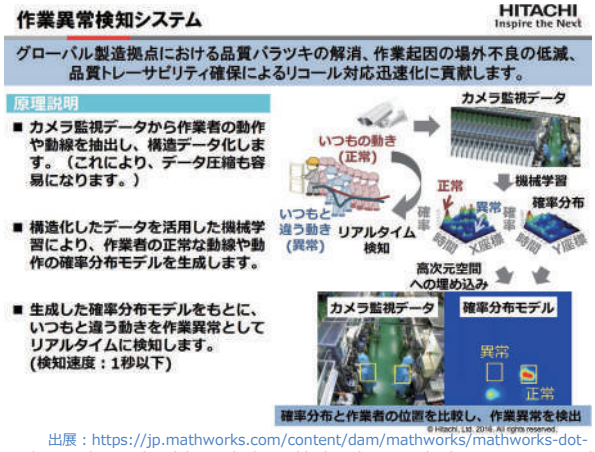
この度、提供を開始する「ワークスタイルの変調」を検知する機能は、従業員ごとの常態化しているワークスタイルと大きく乖離している勤怠行動をAIが自動検知し、それを適切に是正するためのサポートを目的としています。

出展：https://huraid.co.jp/news/20191008/

事例：MATLABによる作業異常検知システムの開発

作業者の動線や動作のデータを基にモデルを構築しているようです。
「通常の動き」と異なる動きを作業者がとった場合、その異常行動を検知していると記載されています。

作業不良の防止と作業者のスキル向上が目的の
取り組みです。



出展：https://jp.mathworks.com/content/dam/mathworks/mathworks-dot-com/images/events/matlabexpo/jp/2016/d1-hitachi-anomaly-detection-using-matlab.pdf

演習

演習1：ホテリング理論による異常検知

- ホテリング理論による異常検知とはどのようか手法か説明してください。
- ホテリング理論の問題点について説明してください。

演習2：LOF法による異常検知

- 局所外れ値因子法（LOF法）による異常検知とはどのような手法か説明してください。

演習3：マハラノビス距離による異常検知

- マハラノビス距離による異常検知とはどのような手法か説明してください。

演習4：スペクトラルクラスタリングによる異常検知

- スペクトラルクラスタリングによる異常検知とはどのような手法か説明してください。

演習5：One Class SVM による異常検知

- One Class SVM による異常検知とはどのような手法か説明してください。

演習6：オートエンコーダーによる異常検知

- オートエンコーダーによる異常検知とはどのような手法か説明してください。

演習7：その他の異常検知アルゴリズム

- 本資料で説明した異常検知アルゴリズム以外に、多くの異常検知アルゴリズムが存在します。どのようなアルゴリズムがあるのか調査してみてください。

第13回：数理最適化

アジェンダ

- 数理最適化とは
- 数理最適化の典型的な問題
- 数理最適化と機械学習

数理最適化とは

数理最適化とは

[Wikipediaより]
数学の計算機科学やオペレーションズリサーチの分野における数理最適化（すうりさいてきか、英: mathematical optimization）とは、（ある条件に関して）最もよい元を、利用可能な集合から選択することをいう。

最も簡単な最適化問題には、ある許された集合から入力をシステマティックに選び、函数の値を計算することによる実数函数の最大化と最小化がある。最適化理論とその手法の、他の形式への一般化は応用数学の広範な分野をなすものである。より一般に、最適化はある与えられた定義域（あるいは制約の集合）についてある目的函数の「利用可能な最も良い」値を見つけることも含む。そのような目的函数と定義域は多様な異なるタイプのものも含む。

数理最適化とは

前ページの定義に記載されていましたが、数理最適化は主に3つの要素で構成されています。

- 定義域（制約の集合）
- 定義域内で変動する変数
- 目的関数

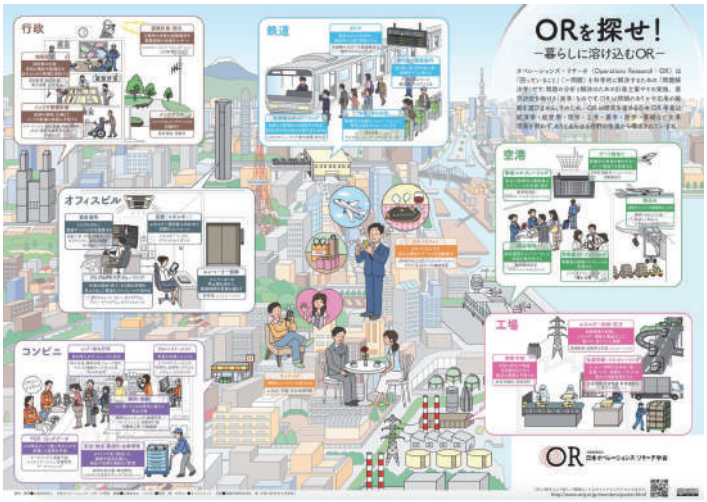
制約のある領域（空間）内において目的関数を最大化する、もしくは最小化など、制約下での最適な解を見つける手法です。

数理最適化の事例

オペレーションズ・リサーチ学会に掲載されているポスターには、様々な事例が記載されています。

配送最適化などは、配送にかかるコストの最小化の事例です。

生産計画・スケジューリングなどは、コストの最小化と収益の最大化の両方の事例だと考えられます。



参照：http://www.orsj.or.jp/members/poster.pdf

問題の分類：線形計画問題

以下の条件を満たす数理最適化の問題を、線形計画問題といいます。

- 目的関数が線形で表される。
- 制約が線形で表される。

例えば、以下のような問題を考えます。

ある人がダイエットをしています。ダイエット中とはいえ、必要な栄養はしっかり取らなければなりません。
そこで、様々なサプリメントを組み合わせて、ダイエット中に必要な栄養をバランス良く取ることを考えました。M種類の栄養素を充足させるために、N種類のサプリメントを摂取します。
各サプリメントごとに摂取できる栄養素の量と価格が異なります。もっとも安上がりにするにはどのサプリメントをどれだけ購入すればよいでしょうか？

問題の分類：線形計画問題

この問題を解くための式は、以下のようになります。
このように目的関数と制約が線形で表現できる問題を、線形計画問題といいます。

$f_{obj} = 10X_1 + 8X_2 + 9X_3$：最小化したい目的関数
X_i：サプリの購入量
$\sum X_i > 0$
$f_{cons1} = 5X_1 + 10X_2 + 8X_3 \geq A_1$：栄養素1の制約
A_1：1日に摂取したい栄養素1の量
$f_{cons2} = 7X_1 + 3X_2 + 8X_3 \geq A_2$：栄養素2の制約
A_2：1日に摂取したい栄養素2の量

	サプリ1	サプリ2	サプリ3
栄養素1[mg]	5	10	8
栄養素2[mg]	7	3	8
価格[円/mg]	10	8	9

問題の分類：非線形計画問題

前述のサプリメントの問題において、価格にボリュームディスカウント（大量に買うと安くなる）が効く場合は目的関数が非線形になります。
このように目的関数や制約が非線形で表現される問題を、非線形計画問題といいます。

$f_{obj} = f_{dis1}(X_1)X_1 + f_{dis2}(X_2)X_2 + f_{dis3}(X_3)X_3$：最小化したい目的関数
X_i：サプリの購入量
$f_{disi}(X_i)$：ボリュームディスカウントの式
$\sum X_i > 0$
$f_{cons1} = 5X_1 + 10X_2 + 8X_3 \geq A_1$：栄養素1の制約
A_1：1日に摂取したい栄養素1の量
$f_{cons2} = 7X_1 + 3X_2 + 8X_3 \geq A_2$：栄養素2の制約
A_2：1日に摂取したい栄養素2の量

	サプリ1	サプリ2	サプリ3
栄養素1[mg]	5	10	8
栄養素2[mg]	7	3	8
価格[円/mg]	10	8	9

数理最適化の典型的な問題

数理最適化の典型的な問題

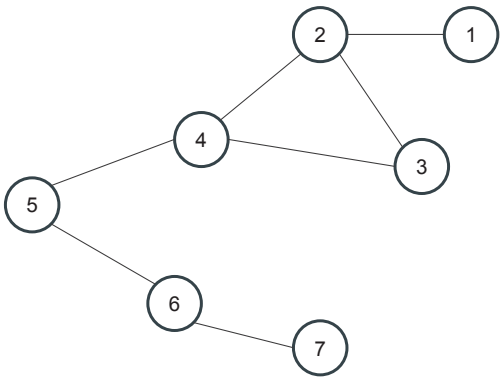
前述のオペレーションズ・リサーチ学会の図にあるように、数理最適化は
様々な場面で応用されています。
それらの中には典型的な問題として定式化されているものがあります。
以降のページでは、その典型例について学習していきます。

典型問題クラス	典型問題
グラフ・ネットワーク問題	最小全域木問題
	最大安定集合問題
	最大カット問題
	最小頂点被覆問題
	最短路問題
	最大流問題
	最小費用流問題
経路問題	運搬経路問題
	巡回セールスマン問題
集合被覆・分割問題	集合被覆問題
	集合分割問題
スケジューリング問題	ジョブショップ問題
	動機スケジューリング問題
切出し・詰込み問題	ナップサック問題
	ビンパッキング問題
	n次元パッキング問題
配置問題	施設配置問題
	容量制約なし施設配置問題
割当・マッチング問題	2次割当問題
	一般化割当問題
	最大マッチング問題
	重みマッチング問題
	安定マッチング問題

参照：https://qiita.com/SaitoTsutomu/items/bfbf4c185ed7004b5721

最短路問題

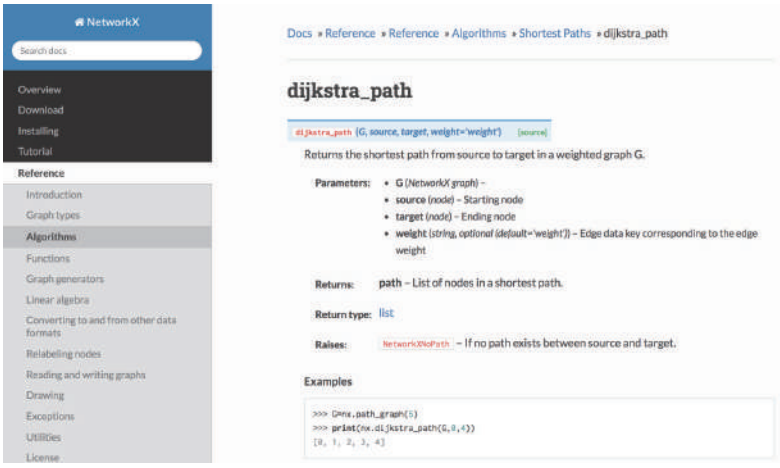
グラフ(ノードとエッジ)で表現できるルートにおいて、特定の場所間で最も短いルートを探索する問題です。
例えば下の図でいうと、ノード2から6までの最短ルートは[2 -> 4 -> 5 -> 6]となります。



最短路問題

[NetworkX]というPythonライブラリを使用すれば、グラフの表現と最短路の探索を簡単に実行することができます。

興味のある方は下記のURLを参考にして実行してみてください。
[https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.shortest_paths.weighted.dijkstra_path.html]

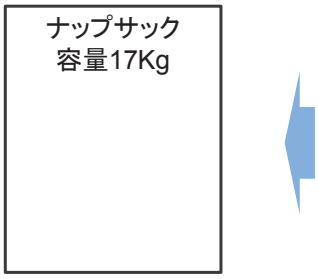


参照：https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.shortest_paths.weighted.dijkstra_path.html

ナップサック問題

ナップサックの中にいくつかの品物を詰め込み、詰め込んだ品物の総価値（価格など）を最大にするという問題です。この問題ではナップサックと品物には容量やサイズが指定されていて、入れた品物のサイズの総和がナップサックの容量を超えてはいけない、という条件があります。

どのように組み合わせて
入れるか

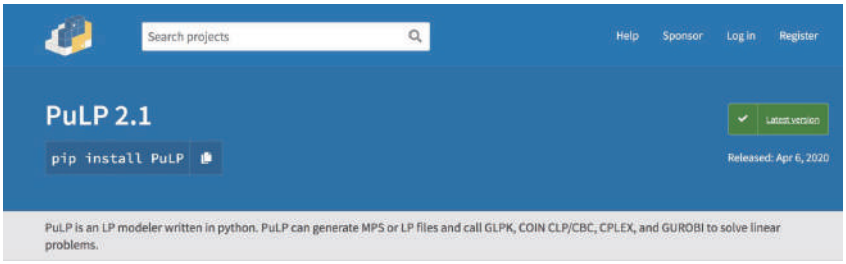


品物	サイズ	価格
1	12Kg	200円
2	5Kg	150円
3	3Kg	100円
4	2Kg	90円
5	7Kg	180円

ナップサック問題

[PuLP]というPythonライブラリを使用すれば、ナップサック問題を解くことができます。

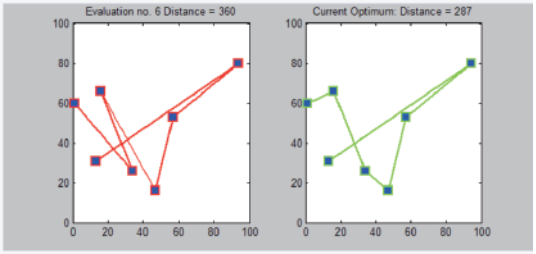
興味のある方は下記のURLを参考にして実行してみてください。
[https://pypi.org/project/PuLP/]



参照: https://pypi.org/project/PuLP/

巡回セールスマン問題

[Wikipediaより]
巡回セールスマン問題(じゅんかいセールスマンもんだい、英: traveling salesman problem、TSP)は、都市の集合と各2都市間の移動コスト(たとえば距離)が与えられたとき、全ての都市をちょうど一度ずつ巡り出発地に戻る巡回路のうちで総移動コストが最小のものを求める(セールスマンが所定の複数の都市を1回だけ巡回する場合の最短経路を求める)組合せ最適化問題である。



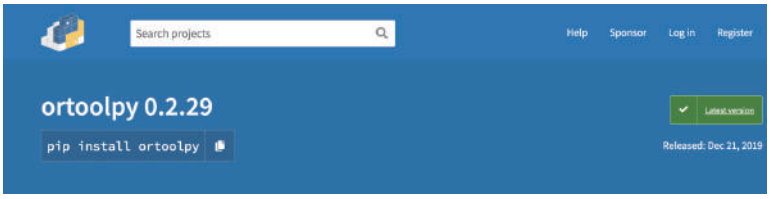
巡回セールスマン問題を総当たりで解く場合のイメージ。左側で一つずつ探していき、より効率のいいルートが見つかった場合、右側のグラフが更新される。

参照: https://ja.wikipedia.org/wiki/巡回セールスマン問題

巡回セールスマン問題

[ortoolpy]というPythonライブラリを使用すれば、巡回セールスマン問題を解くことができます。

興味のある方は下記のURLを参考にして実行してみてください。
[https://pypi.org/project/ortoolpy/]



参照：https://pypi.org/project/ortoolpy/

ポートフォリオ最適化問題

ポートフォリオとは、金融商品の組み合わせのことです。
「ポートフォリオを組む」とは、どの投資信託を購入するか、銘柄で何株ほど持つか、などの検討をすることです。

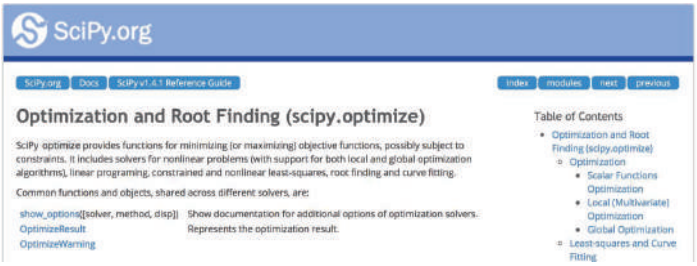
ポートフォリオ最適化問題では、例えば、ポートフォリオが与える収益率の分布（平均と分散）に注目し、ポートフォリオのもたらす収益率の変動の大きさ（リスク）を目的関数に設定します。

また、収益の期待値はX%以上など、制約条件も決めていきます。

ポートフォリオ最適化問題

[Scipy.optimize] を使用すれば、目的関数の最適化（リスクを最小にする）計算が実行できます。
その際に制約条件も考慮することができます。

興味のある方は下記のURLを参考にして実行してみてください。
[https://docs.scipy.org/doc/scipy/reference/optimize.html]
[https://qiita.com/ryoshi81/items/b323a363e5442a15db6d]



参照：https://docs.scipy.org/doc/scipy/reference/optimize.html

数理最適化と機械学習

数理最適化と機械学習

これまでに学習した数理最適化は、目的関数を制約を数式で表現し、数理的に最適解を導くというアプローチでした。

機械学習はデータから何らかのパターンを導き出す手法です。例えばデータの分布をガウス分布と仮定したり、最小化したい損失関数を数式で定義したりと、数理最適化と同じように数理を駆使します。

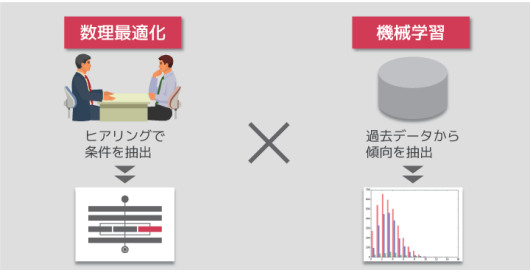
数理最適化と機械学習はそれぞれ使い所があり、それらを組み合わせた事例を紹介します。

数理最適化×ビッグデータ解析で熟練者の暗黙知をデジタルシフト

株式会社日立製作所のHPに掲載されている事例です。
[https://www.hitachi.co.jp/rd/sc/story/mlcp/index.html]

鉄鋼分野における生産計画立案に数理最適化と機械学習を組み合わせたという事例について紹介されています。
現場の熟練者は、業界の深い知識と現場の経験などのドメイン知識に裏付けされた知見を持っています。
それらのドメイン知識は、数理最適化における「XXXなときはYYYしないといけない」などの制約条件に落とし込むことができます。

また、熟練者は過去に膨大な事例を積上げています。
それらの計画履歴を機械学習によってパターン化し、
人はこういうときにはこういう計画を立てやすいという
傾向（特徴量）を抽出するそうです。



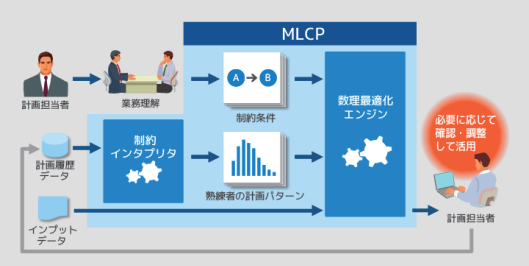
参照：https://www.hitachi.co.jp/rd/sc/story/mlcp/index.html

数理最適化×ビッグデータ解析で熟練者の暗黙知をデジタルシフト

ドメイン知識を落とし込んだ制約条件に、機械学習で抽出した計画の傾向を加えることによって、計画担当者が満足する計画を立てるそうです。

記事の中に、数理最適化と機械学習を融合させた経緯について日立の研究員の方の記述がありました。
「お客さまにすべての条件をヒアリングして、足りない部分があっても聞き出して、全部if thenルール（条件式）で書いていけば、絶対できるという思い込みがありました。しかし、やっぱり限界があって、人の機転や曖昧さというのを全部if thenルールで書くことはできなかった。」

機械学習の長所をうまく融合させた、非常に興味深い事例だと思います。

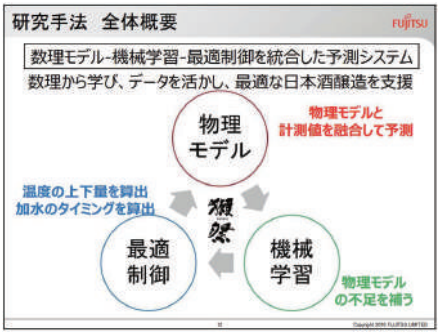


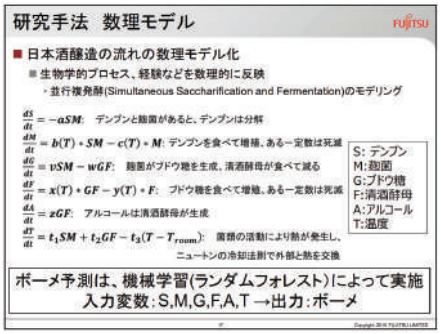
参照：https://www.hitachi.co.jp/rd/sc/story/mlcp/index.html

データ同化と機械学習を用いた実践事例の紹介 ～日本酒醸造AIの実証試験から考える～

東北大学が公開している、株式会社富士通研究所の事例です。
[http://www.ifs.tohoku.ac.jp/edge/DAE/material/DAE5/20190307_KikuchiSama.pdf]

AIを駆使して日本酒醸造の実証実験を行った結果が記載されています。
生物学的プロセスを数理的に表現し、またボーメ（液体の比重を表す単位）の予測を機械学習で実施したそうです。





参照：http://www.ifs.tohoku.ac.jp/edge/DAE/material/DAE5/20190307_KikuchiSama.pdf

演習

演習1：数理最適化の構成

- 数理最適化の主な構成要素3つを挙げてください。

演習2：線形計画問題

- 数理最適化における線形計画問題とはどのような問題か説明してください。

演習3：非線形計画問題

- 数理最適化における非線形計画問題とはどのような問題か説明してください。

演習4：数理最適化の典型問題

- 数理最適化の典型的な問題である以下の問題について、どのような問題であるか説明してください。
  - 最短路問題
  - ナップサック問題
  - 巡回セールスマン問題
  - ポートフォリオ最適化問題

演習5：数理最適化の事例

- 本教材で紹介した数理最適化の事例以外に、どのような事例があるのか調査してみてください。

第14回：自然言語処理

アジェンダ

- 自然言語処理とは
- 自然言語処理の技術
- 自然言語処理におけるDeep Learningの応用

自然言語処理とは

自然言語処理とは

「自然言語」とは何か（Wikipedia：https://ja.wikipedia.org/wiki/自然言語）

- 人間がお互いにコミュニケーションを行うための自然発生的な言語である。「自然言語」に対置される語に「形式言語」「人工言語」がある。形式言語との対比では、その構文や意味が明確に揺るぎなく定められ、利用者に厳格な規則の遵守を強いる（ことが多い）形式言語に対し、話者集団の社会的文脈に沿った曖昧な規則が存在していると考えられるものが自然言語である。自然言語には、規則が曖昧であるがゆえに、話者による規則の解釈の自由度が残されており、話者が直面した状況に応じて規則の解釈を変化させることで、状況を共有する他の話者とのコミュニケーションを継続する事が可能となっている。

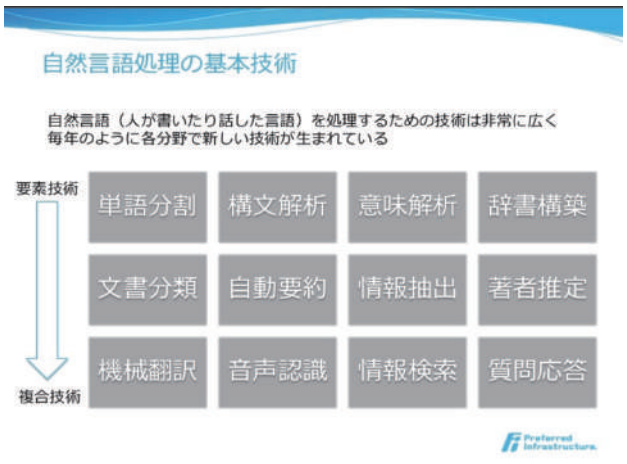
自然言語処理とは

- 「自然言語」は文化圏によって英語/日本語のように表記、文法、発音が異なります。
- 対峙している人同士、場面によって、共通認識を元にしたコミュニケーションが発生するため、全ての情報が自然言語に落ちているわけではなく、解釈が曖昧になることがあります。

- このように複雑な「自然言語」をコンピュータに処理させる技術群のことを自然言語処理といいます。

自然言語処理とは

- 自然言語処理は様々な技術で構成、整理されており、また応用分野も多岐に渡ります。



出展： https://www.slideshare.net/pfi/ss-11474303

自然言語処理の応用

- 検索、翻訳、対話など様々な製品・サービスに自然言語処理が応用されています。
- 以降のページで学習する自然言語処理の構成技術が、これらの製品・サービスでどの様に使われているのか想像してみてください。

出展（Google）

出展： https://www.apple.com/jp/siri/

自然言語処理の技術

単語分割：形態素解析

形態素解析（Wikipedia：https://ja.wikipedia.org/wiki/形態素解析）

- 形態素解析（けいたいそかいせき、Morphological Analysis）とは、文法的な情報の注記の無い自然言語のテキストデータ（文）から、対象言語の文法や、辞書と呼ばれる単語の品詞等の情報にもとづき、形態素（Morpheme, おおまかにいえば、言語で意味を持つ最小単位）の列に分割し、それぞれの形態素の品詞等を判別する作業である。

「おまちしております。」を形態素解析した結果

文字列	読み	原形	品詞の種類	活用の種類	活用形
お待ち	オマチ	お待ち	名詞-サ変接続		
し	シ	する	動詞-自立	サ変・スル	連用形
て	テ	て	助詞-接続助詞		
おり	オリ	おる	動詞-非自立	五段・ラ行	連用形
ます	マス	ます	助動詞	特殊・マス	基本形
。	。	。	記号-句点		

単語分割：N-gram解析

N-gram解析（Wikipedia：https://ja.wikipedia.org/wiki/全文検索）

- 検索対象を単語単位ではなく文字単位で分解し、後続の N−1 文字を含めた状態で出現頻度を求める方法。N の値が1なら「ユニグラム（英: uni-gram）」、2なら「バイグラム（英: bi-gram）」、3なら「トライグラム（英: tri-gram）」と呼ばれる。たとえば「全文検索技術」という文字列の場合、「全文」「文検」「検索」「索技」「技術」「術（終端）」と2文字ずつ分割して索引化を行ってやれば、検索漏れが生じず、辞書の必要も無い。

- 「全文検索技術」という文字列をバイグラムすると「全文」「文検」「検索」「索技」「技術」「術（終端）」と2文字ずつ分割されます。

単語分割：形態素解析とN-gram解析の比較

- N-gramには辞書が不要、という利点はありますが、ノイズが大きいという短所があります（例えば、「東京都」をバイグラムで分割した場合、「東京」と「京都」という結果が含まれてしまいます。）。
- もし検索エンジンにN-gramを採用すると、「東京都の天気」が知りたいのに「東京の天気」と「京都の天気」を検索してしまうかもしれません。
- 機械学習の前処理として形態素解析、N-gram解析を使用する際は、両者の特性を考慮した選択が必要です。

形態素解析とN-gramの比較

	形態素解析	N-gram
インデクシング速度	遅い	速い
インデックスサイズ	小さい	大きい
検索ノイズ	少ない	多い
検索漏れ	多い	少ない
検索速度	速い	遅い
言語依存	辞書が必要	辞書が不要

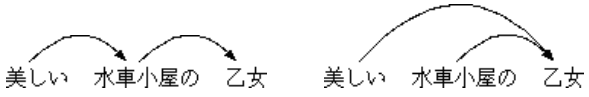
出展： https://ja.wikipedia.org/wiki/全文検索

構文解析

構文解析（Wikipedia：https://ja.wikipedia.org/wiki/構文解析）

- 構文解析（こうぶんかいせき、syntactic analysis あるいは parse）とは、文章、具体的にはマークアップなどの注記の入っていないベタの文字列を、自然言語であれば形態素に切分け、さらにその間の関連（修飾−被修飾など）といったような、統語論的（構文論的）な関係を図式化するなどして明確にする（解析する）手続きである。

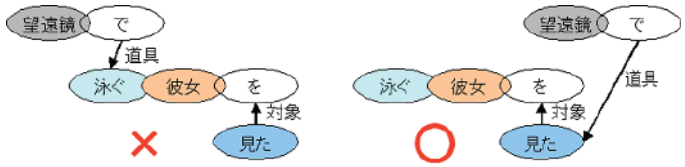
- 「美しい 水車小屋の 乙女」という文章には少なくとも2つの解釈が存在する。「水車小屋が美しい」場合と、「乙女が美しい」場合である。この場合には、意味を含めても正しい解釈がどちらであるか不明であり、その文が置かれた前後の状況、言い換えるとコンテキスト、フレーム情報などを考慮しなければ同定できない。



意味解析

- 下記の例は「望遠鏡で泳ぐ彼女を見た」という文章の解析例です。
- 人間であれば、「望遠鏡を使って泳ぐ人はいない」と判断できるため、すぐに右が正しいと分かります。
- このように、構文解析の結果が「意味」として適当かを判断することを意味解析といいます。

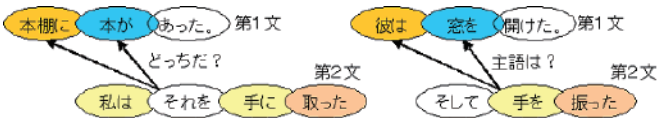
- その他の問題として、「多義性解消」というものがあります。
- 例えば「やった」という言葉には、「宿題をやった（実施した）」のような場合と、「本をやった（与えた）」という解釈の仕方があります。これをどちらか判断することも意味解析といいます。



出展： http://www.sist.ac.jp/~kanakubo/research/natural_language_processing.html

文脈解析

- 構文解析が文単位で行なわれるのに対し、複数の文にまたがる構文木作成＋意味解析を行なうのが文脈解析です。文脈解析は長い文脈に即して行なう必要があるため、単独文の意味解析よりはさらに複雑となります。
- 例えば、「それ」という代名詞が指すのは何か、という問題は文脈解析で解決します。



出展： http://www.sist.ac.jp/~kanakubo/research/natural_language_processing.html

自然言語処理におけるDeep Learningの応用

自然言語処理のタスク

そもそも自然言語処理にはどのようなタスクがあるのでしょうか。
自然言語処理に関する多岐に渡るアルゴリズムのベンチマークを取るために、よく使用されているデータセットがいくつか存在します。

The General Language Understanding Evaluation (GLUE) benchmark
https://openreview.net/pdf?id=rJ4km2R5t7

SQuAD2.0 (The Stanford Question Answering Dataset)
https://rajpurkar.github.io/SQuAD-explorer/

SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference
http://rowanzellers.com/swag/

※ここに示した以外にも多くのデータセットが存在します。興味のある方は「natural language processing datasets」などで検索してみてください。

GLUE

GLUEには9つのコーパス（言語資源）が含まれています。

- CoLA：言語理論に関する書籍や雑誌の記事から引用された英語の受容性の判断で構成されています。各例は文法的な英語文かどうかで注釈が付けられています。
- SST-2：映画の文章で構成されていて、登場人物の感情のレビューと人間による注釈がついています。特定の感情を予測するタスクに使用されます。
- MRPC：オンラインニュースソースから自動的に抽出された文のペアのコーパスです。
- QQP：コミュニティからの質問ペアのコレクションです。タスクは、一対の質問が意味的に同等かどうかを判断することです。

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-B, which is a regression task. MNLI has three classes; all other classification tasks have two. Test sets shown in bold use labels that have never been made public in any form.

出展：https://openreview.net/pdf?id=rJ4km2R5t7

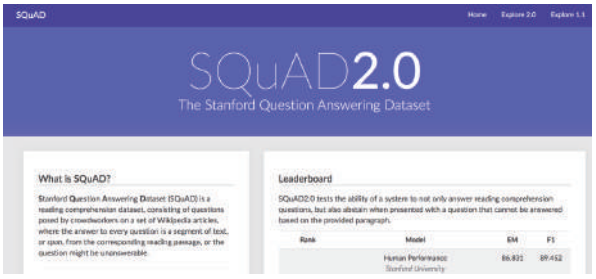
GLUE

- STS-B：文のコレクションです。ニュースの見出し、ビデオと画像のキャプション、および自然言語推論データから抽出されたペアデータです。
- MNLI：テキストの含意アノテーションが付いた文のコレクションです。タスクは、前提が仮説を伴うか（含意）、仮説と矛盾するか（矛盾）、またはどちらともいえない（中立）かを予測することです。
- QNLI：質問応答データセットです。タスクは、文脈文に質問への答えが含まれているかどうかを判別することです。
- RTE：MNLIと同じ用に、テキストの含意アノテーションが付いた文のコレクションです。
- WNLI：読解タスクです。代名詞付きの文を読み、その代名詞の指示対象を選択肢のリストから選びます。

SQuAD

ウィキペディアの一連の記事に対し、クラウドワーカーが提起した質問と回答で構成される読解データセットです。

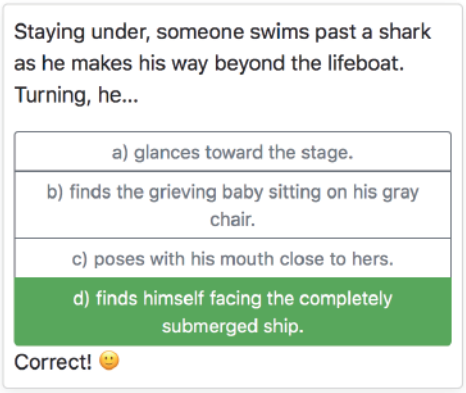

```
{
 "question": "When did Beyonce start becoming popular?",
 "id": "56be85543aeaaa14008c9063",
 "answers": [{"text": "in the late 1990s", "answer_start": 269}],
 "is_impossible": false
},

{
 "question": "What is the original meaning of the word Norman?",
 "id": "56dde0379a695914005b9636",
 "answers":
  [
   {"text": "Viking", "answer_start": 341},
   {"text": "Norseman, Viking", "answer_start": 331},
   {"text": "Norseman, Viking", "answer_start": 331}
  ],
 "is_impossible": false
}
```

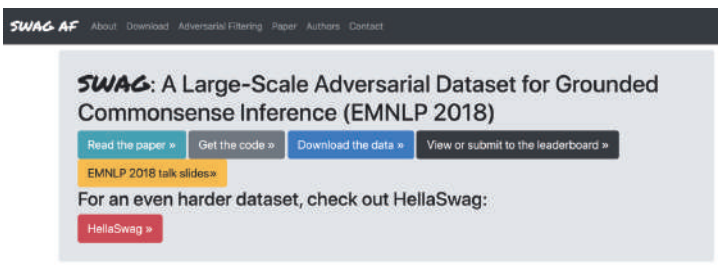



出展：https://rajpurkar.github.io/SQuAD-explorer/

SWAG

推論タスクのための大規模なデータセットです。
入力文に続く文を、4つの選択肢から選びます。





出展：http://rowanzellers.com/swag/

TransformerとBERT

これまで自然言語処理に関する多岐に渡るアルゴリズムのベンチマークを取るために、よく使用されているデータセットについて紹介してきました。
これらのデータセット使ったベンチマークで、過去のアルゴリズムより優れているとして2018年ごろから登場してきたのがBERTです。

この技術の何が優れているのか、TransformerとBERTというキーワードを基にご紹介します。

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

出展：https://arxiv.org/pdf/1810.04805.pdf

Transformerとは

2017年にGoogleとトロント大学から「Attention Is All You Need」という論文が発表されました。

論文には「We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.」と書かれており、これまで自然言語処理として活用されていた再帰型ニューラルネットワーク（RNN）や畳込みニューラルネットワーク（CNN）を使用せず、「attention mechanisms」だけによるシンプルな機構を
提案する、とされています。


Attention Is All You Need

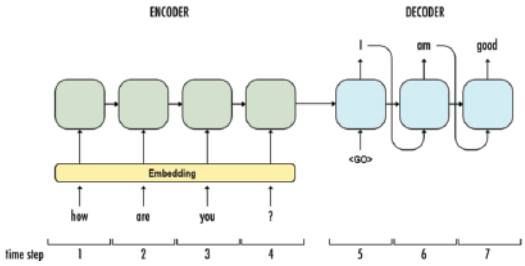
Abstract

出展： https://arxiv.org/pdf/1706.03762.pdf

Transformerとは

例えば翻訳や対話を扱うモデルを考えます。
入力文と出力文は、長さが異なることが一般的です。
しかし機械学習モデルは固定長で扱うことが要求されるため、Encoder-Decoderモデルが考え出されました。

また、単語の出現の順番を考慮する場合はRNNが活用されてきましたが、固定長の問題しか扱えませんでしたし、
文が長くなる場合は勾配計算が困難になるという問題もありました。



$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i \mid w_{i-k} \dots w_{i-1})$$

出展： https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129

Transformerとは

Transformerは、過去の自然言語処理で多く使われるRNNやCNNを「Self-Attention Layer」に入れ替えたモデルです。

Attention(注意機構)とは、単語同士の関係を行列で表す方法です。
行列で表現するので、RNNと比べて並列計算が可能となります。

CNNでの自然言語処理はカーネルで単語関係を見るため、計算量を削減するためにカーネルサイズを小さくすると、
長文における単語の前後関係が表現できなくなります。
Attentionは、CNNと比べて長文の為の深いモデル構築が不要となります。

このAttentionとは一体何なのか、以降のページで解説していきます。

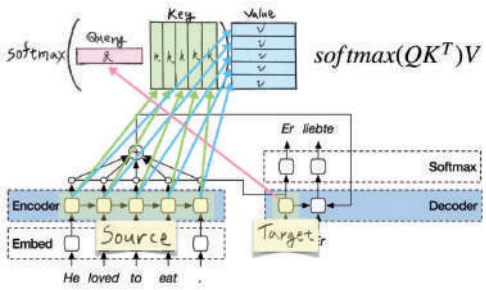
Attentionとは

EncoderとDecoderを使うことの目的は、「重要な情報を残す」ということです。
「EncodeしたものをDecodeしたら、Encode前と近い情報に復元できた」という場合、重要な情報を抽出できたということです。

AttentionはEncoderの隠れ層とDecoderの隠れ層の値を使って、以下のように計算します。
Encoderの隠れ層はkeyとvalueになります。Decoderの隠れ層はqueryになります。

まずはじめにqueryとkeyの内積を取ります。ここでqueryとkeyの類似度を測ります。内積を取った後はsoftmax関数に掛けるので、その値はqueryに一致したkeyの位置を表現することになります。

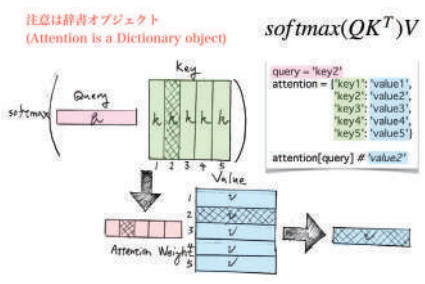
最後に、softmax関数の出力値とvalueの内積を取ります。
これはkeyの位置に対応するvalueを加重和として取り出す操作です。

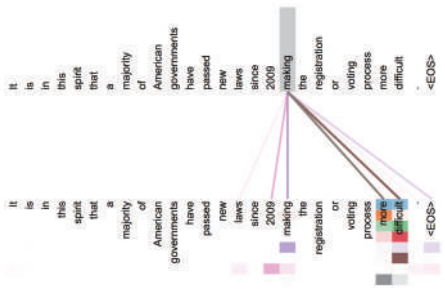


出展：http://deeplearning.hatenablog.com/entry/transformer

Attentionとは

つまり、attentionとは queryに一致するkeyを索引し、対応するvalueを取り出す操作で、辞書オブジェクトと同じ機能です（左下図）。
入力文の全範囲が埋め込まれたEncoder隠れ層とDecoder隠れ層に渡ってこのような操作を実行することで、モデルが構文や意味構造を表現できるようになることが報告されています。
右下図の上段はquery、下段はvalueです。8色のマーカーの濃淡はattentionの重みの大きさです。
この例では「making」の長期依存を捉え、「making...more difficult」という句を形成しています。



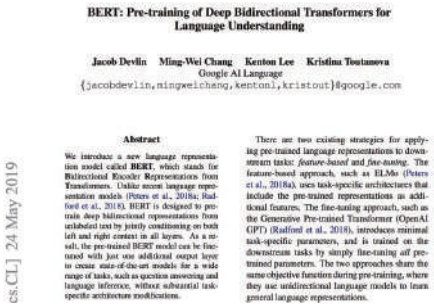


出展：http://deeplearning.hatenablog.com/entry/transformer

BERTとは

2019年に採択されたGoogleによる論文でBERTが紹介されています。
「We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers.」
とあり、Transformerを使っていることが記載されています。

「Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.」
より、Transformerをラベルのないテキストで事前トレーニングするように設計されているそうです。その際、すべての層で左と右の両方のコンテキスト（文脈）を組み合わせて調整しているそうです。



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

cs.CL] 24 May 2019

Abstract

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

出展：https://arxiv.org/pdf/1810.04805.pdf

BERTとは

Attentionを活用したTransformerを使っているBERTは並列計算ができるため、計算資源を有効に活用することができます。また、多くの情報を詰め込むことができるため、過去に報告されてきた自然言語処理のモデルよりも高い精度を達成しています。

下の表はGLUEテストにおいて、BERTが他の手法よりも精度が高かったという報告です。

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

出展：https://arxiv.org/pdf/1810.04805.pdf

参考記事

更に学習されたい方は、以下の記事などを参考にされてください。

Attention Is All You Need
https://arxiv.org/pdf/1706.03762.pdf

論文解説 Attention Is All You Need (Transformer)
http://deeplearning.hatenablog.com/entry/transformer

自然言語処理の巨獣「Transformer」のSelf-Attention Layer紹介
https://medium.com/lsc-psd/自然言語処理の巨獣-Transformer-のSelf-Attention-Layer紹介-a04dc999efc5

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
https://arxiv.org/pdf/1810.04805.pdf

演習

演習1：自然言語処理とは

- 自然言語処理とはなにか、説明してください。
- 自然言語処理を構成する技術にはどのような技術があるのか、説明してください。

演習2：自然言語処理の発展

- 自然言語処理にDeep Learningが活用されるようになり進歩した技術は何でしょうか。調査し、要点をまとめてください。

演習3：AttentionとTransformer

- AttentionやTransformerは、それまでのDeep Learningを活用したモデルと比べて、何が優れているのでしょうか。コンピュータの計算資源、モデルの表現力などの視点から要点をまとめてください。

第15回：グラフ理論

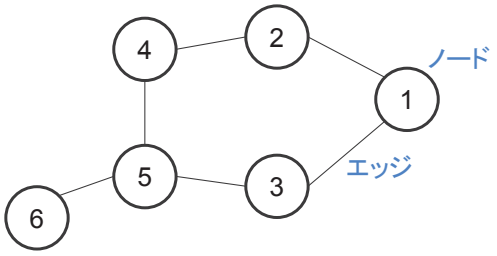
アジェンダ

- グラフ理論
- グラフの指標
- グラフ理論と機械学習
- グラフ理論の応用事例

グラフ理論

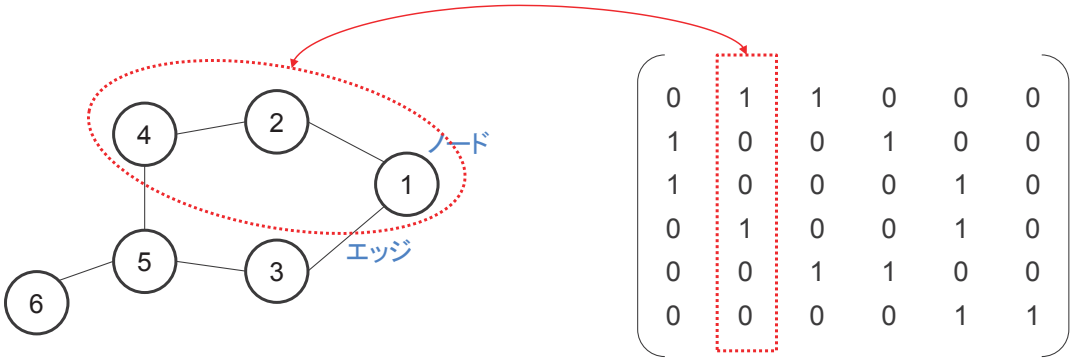
グラフ理論とは

グラフ理論(Graph theory)とは、ノード(節点・頂点)の集合とエッジ(枝・辺)の集合で構成されるグラフに関する数学の理論です。



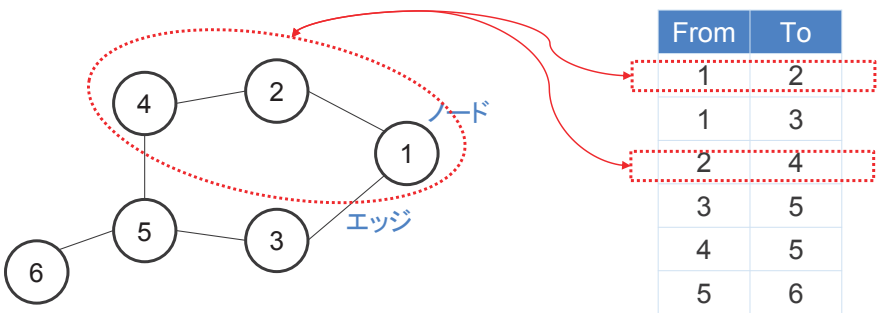
グラフ理論の表現方法

グラフの表現には、図ではなく行列を使う方法もあります。
下の例では、ノードの数6を辺とする6×6の行列で表現しています。



グラフ理論の表現方法

グラフを行列で表現した場合、ノードの数がNだとすると、必ずN^2の情報を保持することになります。
これでは、分析の際にコンピュータのリソースが枯渇してしまいます。
つながり（エッジ）が存在しないノードは無視し、つながりが存在するノードのみの情報で表現する方法もあります。

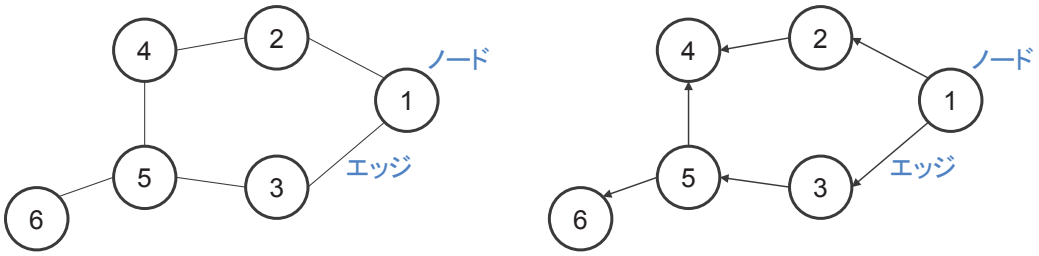


From	To
1	2
1	3
2	4
3	5
4	5
5	6

※From / Toと記載されていますが、エッジには方向がない無向グラフを例にしているため、[From:1, To:2]と[From:2, To:1]は同じ意味になります。

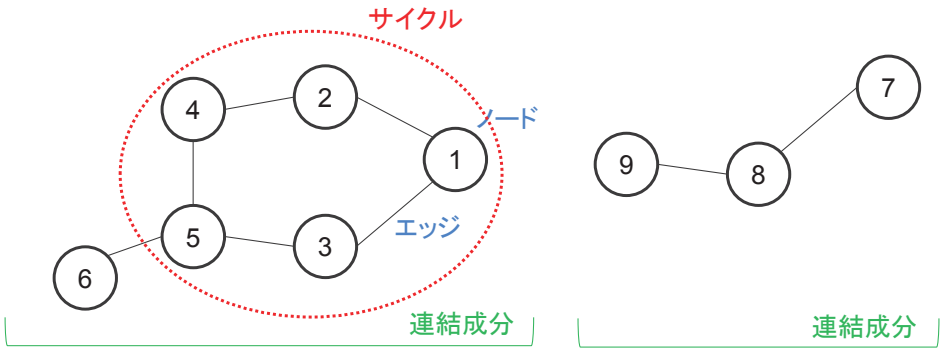
無向グラフと有向グラフ

エッジに向きがないグラフ(下左図)を無向グラフといいます。
エッジに向きがあるグラフ(下右図)を有向グラフといいます。



サイクル、連結成分

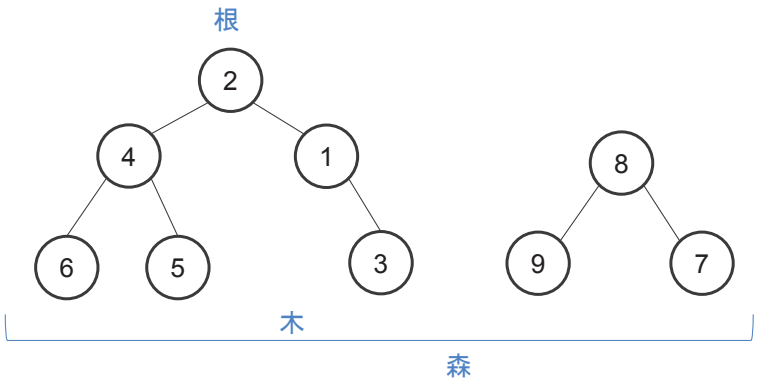
ノードとエッジを辿ると元に戻ってくる経路をサイクルといいます。
パスがつながっているかたまりを連結成分といいます。



木

連結でサイクルがないグラフを木といいます。
木が複数あるものを森といいます。

木は、あるノードを根として、そこからグラフがはじまると考えることがあります。



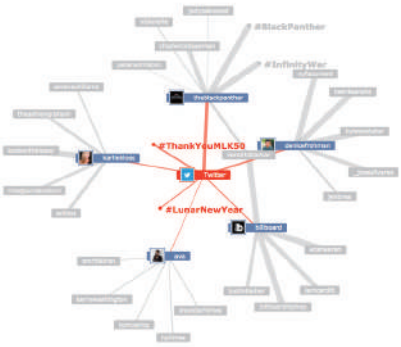
グラフの指標

グラフをどのように評価するか

右図はTwitterにおけるユーザー間のつながりを可視化するツールです。

誰かを中心としたグループが見られたり、関係の近い人/遠い人が見られたりします。
このようなグラフを分析すると、「誰が影響力の強い人なのか？」、「Aさんと知り合いになるためには誰とつながればいいのか？」などの分析ができるようになります。

以降のページでは、グラフを評価する際に用いられる各種指標について解説していきます。



参照： https://mentionmapp.com/#

中心性

次数中心性

- 次数とはノードに接続しているエッジ数のことです。「次数中心性」とはエッジが多いノードを高く評価する指標です。

固有ベクトル中心性

- 注目するノードに対してエッジを張っているノードが、どれだけの中心性を持っているかという指標です。注目するノードにつながっているノードが、どれくらい影響力のあるノードなのかという見方ができます。

媒介中心性

- 任意の2つのノードの最短経路に、注目するノードが含まれる確率のことです。ネットワークグラフにおいて、注目ノードがどれくらい経由されているか、という指標になります。

中心性

PageRank

- 次の3つの考え方を基礎とした指標です。Webサイト間の重要度を測るために使用されています。（1）ノードiの中心性は沢山のノードからリンクが貼られている程（=次数が高い程）高くなる。（2）ノードiがリンクしているノードjの中心性が高い程、ノードiの中心性も高くなる。　（3）ノードiがリンクしているノードjの中心性の価値は、ノードjの次数が低い程高くなる。

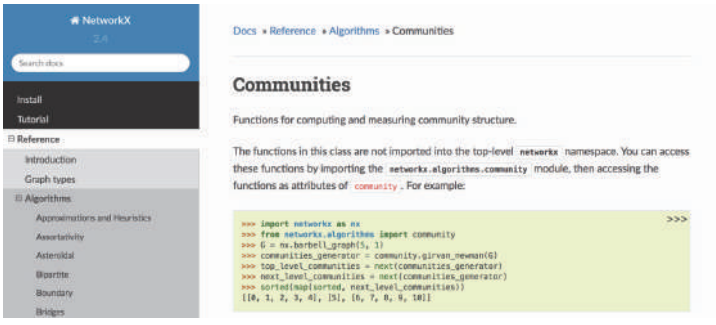
近接中心性

- 注目するノードから他のノードまで、平均的にどれくらいの近さかを表す指標です。

コミュニティ

グラフをいくつかの「コミュニティらしい」グループに分割し、コミュニティ内で似通ったこと現象がないか、などの分析につなげることがあります。

コミュニティに分割には様々な手法があり、Pythonのライブラリを使って実行することができます。
興味のある方は以下のURLを参考に試してみてください。



参照：https://networkx.github.io/documentation/stable/reference/algorithms/community.html

演習1：グラフ指標の算出とコミュニティ分割

- [演習/Chapter15_graph_theory.ipynb]を実行し、グラフ理論における各種指標の算出、コミュニティの分割について確認してください。

グラフ理論と機械学習

グラフデータを説明変数に使うには

例えばアヤメデータを考えます。
データは2次元の表形式ですので、sepal_length/sepal_width/petal_length/petal_widthをそのまま説明変数として使用することができます。

アヤメデータ

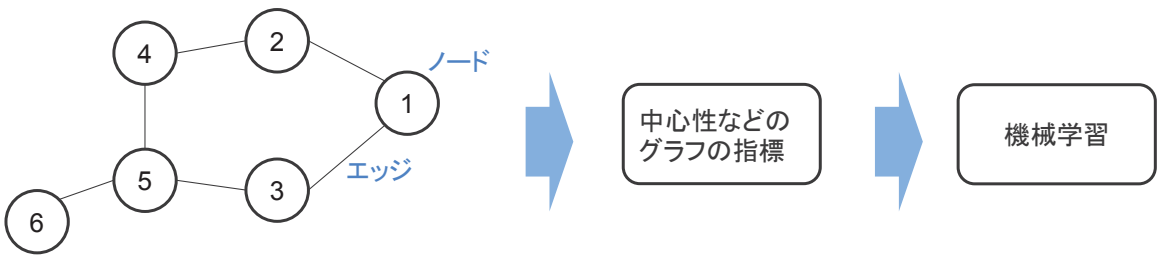
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

花弁とがく片の長さ、幅のデータ。
あやめの種類を判定するための説明変数として使用する。

あやめの種類。
目的変数として使用する。

グラフデータを説明変数に使うには

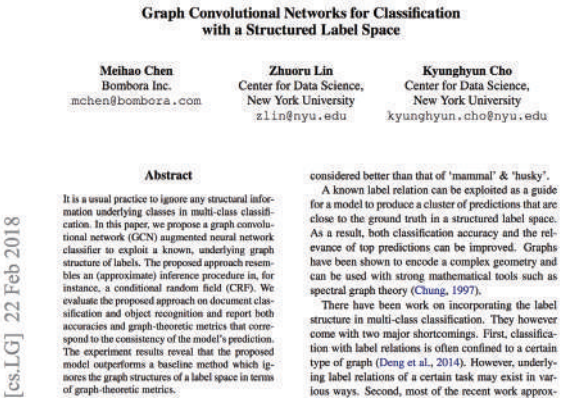
グラフはノードとエッジ、エッジの方向性などを持つデータですので、そのままでは機械学習の説明変数としては使えません。そこで、いったんグラフの指標に変換し、説明変数として使用することがあります。



グラフデータを説明変数に使うには

グラフを指標に直してから機械学習に投入する、というプロセスを経ることなく、直接ニューラルネットに投入してしまおうというアイディアも出てきています。

「グラフ畳み込み」と言われる手法で、Kerasによる実装も公開されていますので、興味のあるかたは実行してみてください。
https://github.com/tkipf/keras-gcn



Graph Convolutional Networks for Classification with a Structured Label Space

Meihao Chen
Bombora Inc.
mchen@bombora.com

Zhuoru Lin
Center for Data Science,
New York University
zlin@nyu.edu

Kyunghyun Cho
Center for Data Science,
New York University
kyunghyun.cho@nyu.edu

Abstract

It is a usual practice to ignore any structural information underlying classes in multi-class classification. In this paper, we propose a graph convolutional network (GCN) augmented neural network classifier to exploit a known, underlying graph structure of labels. The proposed approach resembles an (approximate) inference procedure in, for instance, a conditional random field (CRF). We evaluate the proposed approach on document classification and object recognition and report both accuracies and graph-theoretic metrics that correspond to the consistency of the model's prediction. The experiment results reveal that the proposed model outperforms a baseline method which ignores the graph structures of a label space in terms of graph-theoretic metrics.

considered better than that of 'mammal' & 'husky'.

A known label relation can be exploited as a guide for a model to produce a cluster of predictions that are close to the ground truth in a structured label space. As a result, both classification accuracy and the relevance of top predictions can be improved. Graphs have been shown to encode a complex geometry and can be used with strong mathematical tools such as spectral graph theory (Chung, 1997).

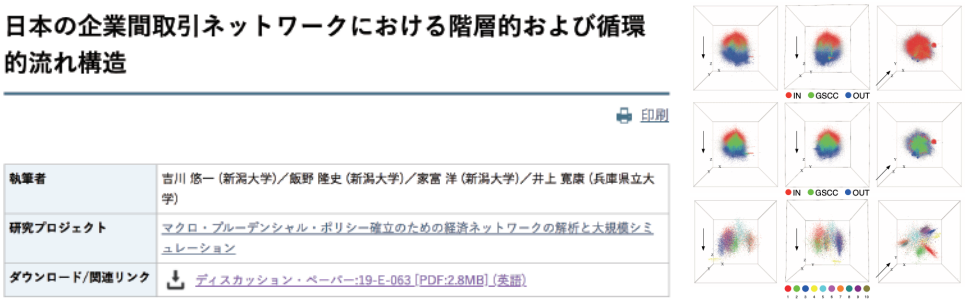
There have been work on incorporating the label structure in multi-class classification. They however come with two major shortcomings. First, classification with label relations is often confined to a certain type of graph (Deng et al., 2014). However, underlying label relations of a certain task may exist in various ways. Second, most of the recent work approx-

参照：https://arxiv.org/pdf/1710.04908.pdf

グラフ理論の応用事例

企業間取引の分析

[サイトより]企業間に働く種々の相互作用は、景気変動やインフレ/デフレをはじめとしてさまざまなマクロ経済現象において重要な役割を果たすと考えられている。例えば、各企業は直接あるいは間接的に商取引関係によって互いにつながり、複雑な生産ネットワークを形成している。東京商工リサーチ社により収集された企業間取引関係のデータ(2016年)は、日本における約100万の企業間の約500万におよぶ取引関係を収録した大規模データである。著者らは、RIETIによって提供されたこの貴重なデータを用いて、企業レベルでわが国の生産ネットワークの構造について実証的研究を行なっている。



日本の企業間取引ネットワークにおける階層的および循環的流れ構造

印刷

執筆者	吉川 悠一 (新潟大学)／飯野 隆史 (新潟大学)／家富 洋 (新潟大学)／井上 寛康 (兵庫県立大学)
研究プロジェクト	マクロ・プルーデンシャル・ポリシー確立のための経済ネットワークの解析と大規模シミュレーション
ダウンロード/関連リンク	ディスカッション・ペーパー:19-E-063 [PDF:2.8MB] (英語)

参照：https://www.rieti.go.jp/jp/publications/nts/19e063.html

メルペイの不正決済検知への機械学習活用
グラフ理論で“疑わしい人”を事前に推定する

メルペイでの不正決済検知への機械学習の活用事例と，グラフ理論を活用した実験について紹介されています。[サイトより]メルカリでお客様の取り引きがたくさんあったら膨大なグラフができると思うんですけど、それに対して予め僕らがこの辺の人たちがきっと怪しいという印を事前に付けておく。そこで実際に特定の取り引きが行われたときに、ビビビッとアラートが鳴ってそこで検知する。そうするとその人たちが一気にガバッと取るという、すごくシンプルな話なんですけど、そういうことをやりたいというのがアイディアです。

Overview

全体の膨大な取引の中から，疑わしい人たちのつながりを，

事前に予測していくつかの小さなグラフ(Sub Graph)を作っておき，

その Sub Graph 内のアカウント間で疑わしい取引が発生したら検知をする

というアイデア．

参照：https://logmi.jp/tech/articles/322735

令和２年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

令和２年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

人工知能特論

令和３年２月

学校法人電子学園（日本電子専門学校）
〒169-8522　東京都新宿区百人町 1-25-4
TEL 03-3369-9333　FAX 03-3363-7685