第 5 章

構造化データ処理の基本

構造化データとはどのようなものかを正しく理解し、データ処理を行う際の基本構文を学びましょう。

この章のゴール

構造化データの基本集計、処理を行えるようにする。

構造化データ処理の基本

構造化データの処理の基本では、いわゆる前処理の必要性を説明します。取得したデータにはかならず不要なデータが混じっていると考えるほうが良いでしょう。そのため、最初に前処理と言われる作業が必要になります。データの重複や欠損、本来必要ではないデータの混入、異常値や外れ値、このようなデータを抽出し、精査しなくてはなりません。また、どんなデータなのかを把握し、分析するのに扱いやすい形に整形することはとても重要な作業となります。

また、「第 1 章:データサイエンス(機械学習のアルゴリズム)によるデータ解析が社会にもたらす変化　データとはなにか」で示したとおり、構造化データとは、行と列の 2 次元のデータ、または表形式に変換可能なデータです。構造化データは、規則的で分かりやすい構造でリレーショナルデータベースに用いられます。

リレーショナルデータベース(Relational Database:RDB)は、現在最も広く構造化データ処理の代表格として利用されているデータベースです。

構造化データで管理するため、理解しやすいという特徴があります。

テーブルに格納される値は主に数値や文字で表現され、ほとんどの場合、行(レコード・ロウ)ごとに顧客や商品など、人やモノの情報を管理し、列(カラム)ごとに同じ意味合いの項目データが格納されます。また、リレーショナルデータベース内の 1 テーブルのカラムは、1 フィールドと決まっており、1 テーブル内のカラムに 2 フィールド割り当てられることはありません。また、リレーショナルデータベースは、主に SQL(Structured Query Language)という専用の言語を用いてデータを操作します。

並べ替え

並べ替えは、主にテーブルのカラム別の昇順・降順にデータを並べ替えることを指し、sort と言います。構造化データは、行と列の 2 次元のデータですのでカラムごとに昇順に並べたり、降順に並べかえることが安易に出来ます。

コンピュータ処理をするにも並べ替えを行うことによって、その後の処理速度が高くなります。

また、取得したデータにはかならず不要なデータが混じっていると考える方が良いため、データを並べ替えることで、重複や欠損、データの異常値等を目視でも確認しやすくすることができます。

抽出

抽出とは、取得したデータから目的に応じて必要な条件を指定し、データを取り出すことを指します。また、一般的に、取得したデータをそのまま使用することはできませんので取得したデータから、必要なデータを必要な分だけ抽出し、使用できるデータに変換、集計、結合して分析の目的に合ったデータのみを使用するデータクレンジングを目的として行うことが出来ます。

データを抽出して使用することで、データ規模を縮小することができ、処理速度も高くなります。

データ型変換

データ型には、数値として認識する、数値データ、テキストとして認識する文字データ（テキストデータ）、年月日等時間として認識する、日時データと、様々な細かい分類があります。フィードに、適切なデータ型が指定されていない場合では、正しく認識されないため、結果の取得が困難だったり、文字化けの原因になります。
カラム内のデータがどのようなデータなのかを確認し、適切なデータ型を設定する必要があります。適切なデータ型ではない場合はデータ型の変換が必要になります。

データを取得し、使用するツールにそのデータを読み込んだ場合、データ作成時のデータ型が自動的に登録される場合が多いです。ですが、実際に、データ定義とそのデータ型が合っているかの確認は必須となります。

集計

分析するためには集計データは必須です。分析の前処理段階としてデータ内容確認 （レコード数／カラム数）や、基本統計量算出、また、破損／欠損がどのくらいあるか等、取得データ把握のために全て集計する必要があります。

集計値を確認することで、データ内の不備や矛盾等も見つけ出すこともできます。

取得データの把握時点でも、集計は大切な役割を担っています。

よく、集計は食材、分析は調理法という例えを耳にします。おいしい料理を食べるためには、どんなに調理法だけが良くてもおいしい料理は出来上がりません。むしろ食材が良ければ極端な話そのままでもある程度、おいしくいただくことは出来るのです。

重複・欠損・異常値の処理

重複とは、データ内のレコード単位で一意ではないデータがある場合等、2 つ以上データが重なることを言います。しかし、場合によってはそのまま使用できる、使用する場合もありますので、重複処理をする場合には、取得データの取得状況や目的に応じて、どういう範囲でどういうものを「重複データ」とするかが重要となります。どのような場合を「重複データ」とするのかをしっかり把握することができたら、目的に応じてデータの削除や補完もしくは修正等、重複データについての処理を進めます。

欠損値とは、分析に利用するデータにおいて、何らかの理由によりデータが記録されず存在していない状態を言います。欠測、欠落とよばれることもあり、英語では missing data といいます。データの欠損状況は、分析用のデータセットが揃った時点で

1. サンプル単位での欠損状況の確認（レコード・行単位）
2. 変数内での欠損状況の確認（カラム・列単位）

を行います。これらの過程で欠損値があった場合は以下の様に対処いたします。

●レコードごと除く

欠けた変数が多いレコードをそもそも削除して対処することは非常にシンプルで理解しやすい対処です。特に、全データのうち数件しかないという場合であれば、この対処で問題ありません。

では、分析しようとする対象のデータの多くに欠損があるという場合ではどうすればいいでしょうか。分析対象は 1 万件あるのに、半数近くが欠損ばかりという状況では、適切な分析は実行できるでしょうか？ このとき

「どのような情報が欠損しているのか」「欠損している情報は分析課題を解くために必要なのか」という視点が非常に大切になります。

●欠損の多い変数を除く

分析課題を解くために必要でない変数は、分析の対象外として差し支えないと言えます。ただし、欠損している理由や背景については十分に確認することが求められます。私達の手元には、何かが起きた「結果」としてデータがやってきます。そのため「欠損が多く起こっている」ということも情報となります。システムにおけるバグやエラー、抽出やデータ化時の人的ミスなど「欠損が起きている状況」の把握は、実は新たなインサイトの発見のきっかけとなることが多くあります。

●補完する

欠損値の補完には多くの方法がありますが、シンプルでよく知られた方法として、平均値や最頻値等の代表値で補完するという方法があります。ただし闇雲に代表値を入れればいいというものではもちろんありません。

その極端に小さな値、あるいは極端に大きな値を言います。それら「外れ値」の中でも、外れている理由が判明しているものが「異常値」です。データ取得の状況を把握していれば、異常値が必要なデータなのかそうでないのかは判断できます。しかし、もしこのデータの取得背景がわからなければ慎重に扱う必要があります。また、数値データの中に数字ではなく文字列や記号などが入っているケースもあるでしょう。　これらは異常値とは呼ばず、ノイズと呼びます。外れ値と異常値はこのように異なるものですが、英語では同じ「outlier」と言います。

データの結合

業務システムのデータベースは、通常、データの種類ごとにテーブルが分かれているため、必要なデータが 1 つのテーブルにすべて入ったデータを取得できることはとても珍しいことです。しかし、分析用のデータとしては、1 つのテーブルに必要な情報が全てまとまっているデータの方が望ましく、そのようなデータを得るためには、データを取得した後に、テーブル同士の結合処理が必要になってくる場合があります。

データの結合は、抽出する条件でレコードを絞り込み、必要なレコードのみを結合します。そうすることで、その後の処理で、メモリや CPU 等のリソース消費や、コンピュータの処理時間を抑えることができます。