

# Package datadepot

**Description** The **datadepot** package provides a collection of datasets used in the book Data Science Foundations and Machine Learning with Python.

**URL** <https://github.com/vanraak/datadepot>

**Depends** Python ( $\geq 3.8$ ) and Pandas ( $>2.0$ )

**License** GPL ( $\geq 2$ )

**Repository** Pypi

**Authors** Jeroen van Raak

**Maintainer** Jeroen van Raak, [j.j.f.vanraak@uva.nl](mailto:j.j.f.vanraak@uva.nl)

## Installation

```
pip install datadepot
```

or

```
conda install conda-forge::datadepot
```

## Usage

```
import datadepot
df=datadepot.load("<dataset>")
```

Replace with the name of the dataset, such as “bank”, “house”, or “churn”.

## Example

```
df=datadepot.load("bank") # Load the bank dataset.
```

## Datasets

adult	3
bank	5
cereal	7
churn	9
churn_ibm	11
churn_mlc	13
covid	15
credit	16
credit_card	17
cpu	18
diamonds	19
drug	20
gapminder	21
hotel_city	22
hotel_resort	25
house	28
house_price	29
insurance	30
las_vegas	31
loan	33
machine_failure	34
mpg	35

<b>red_wines</b>	<b>36</b>
<b>vehicle</b>	<b>38</b>
<b>white_wines</b>	<b>39</b>
<b>wholesale_wines</b>	<b>41</b>

---

## adult Adult dataset

---

### Description

The Adult dataset was collected by the US Census Bureau. The primary task is to predict whether a given adult makes more than \$50K per year based on attributes such as education, hours worked per week, and other demographic features. The target variable is income, a factor with levels “<=50K” and “>50K”, and the remaining 14 variables are predictors.

### Usage

```
df=datadepot.load(“adult”)
```

### Format

The adult dataset contains 48,598 rows and 14 columns.

The 14 variables are:

- age: age in years.
- workclass: a factor with 6 levels.
- demogweight: the demographics to describe a person.
- education: a factor indicating the highest level of education attained (16 levels).
- education\_num: ordinal encoding of education level.
- marital\_status: a factor with 5 levels.
- occupation: a factor with 15 levels.
- relationship: a factor with 6 levels.
- race: a factor with 5 levels.
- gender: a factor with levels “Female”, “Male”.
- capital\_gain: capital gains.
- capital\_loss: capital losses.
- hours\_per\_week: number of hours of work per week.
- native\_country: a factor with 42 levels.
- income: yearly income as a factor with levels “<=50K” and “>50K”.

### Source

This dataset is publicly available from UCI Machine Learning Repository: <https://archive.ics.uci.edu/dataset/2/adult>

### Reference

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. Kdd.

### Notes

We changed some of the column names, dropped the “education\_num” column, and fixed typos in the “native\_country” column.

---

## bank Bank Marketing dataset

---

**Description** This dataset contains data from direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed. The classification goal is to predict if the client will subscribe a term deposit (variable deposit).

### Usage

```
df=datadepot.load("bank")
```

### Format

The bank dataset contains 4,521 rows (customers) and 17 columns. The 17 variables are explained below.

Bank client data:

- age: numeric.
- job: type of job; categorical: "admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services".
- marital: marital status; categorical: "married", "divorced", "single"; note: "divorced" means divorced or widowed.
- education: categorical: "secondary", "primary", "tertiary", "unknown".
- default: has credit in default?; binary: "yes", "no".
- balance: average yearly balance, in euros; numeric.
- housing: has housing loan? binary: "yes", "no".
- loan: has personal loan? binary: "yes", "no".

Related with the last contact of the current campaign:

- contact: contact communication type; categorical: "unknown", "telephone", "cellular".
- day: last contact day of the month; numeric.
- month: last contact month of year; categorical: "jan", "feb", "mar", ..., "nov", "dec".
- duration: last contact duration, in seconds; numeric.

Other attributes:

- campaign: number of contacts performed during this campaign and for this client; numeric, includes last contact.
- pdays: number of days that passed by after the client was last contacted from a previous campaign; numeric, -1 means client was not previously contacted.

- previous: number of contacts performed before this campaign and for this client; numeric.
- poutcome: outcome of the previous marketing campaign; categorical: “success”, “failure”, “unknown”, “other”.

Target variable:

- deposit: Indicator of whether the client subscribed a term deposit; binary: “yes” or “no”.

### **Source**

This dataset is publicly available from UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/bank+marketing>

### **References**

Moro, S., Laureano, R. and Cortez, P. (2011) Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference.

---

## **cereal** Cereal dataset

---

### **Description**

This dataset contains nutrition information for 77 breakfast cereals and includes 15 variables.

### **Usage**

```
df=datadepot.load("cereal")
```

### **Format**

The cereal dataset contains 77 rows (breakfast cereals) and 15 columns.

The 16 variables are:

- name: Name of cereal.
- manuf: Manufacturer of cereal:
  - A: American Home Food Products;
  - G: General Mills;
  - K: Kelloggs;
  - N: Nabisco;
  - P: Post;
  - Q: Quaker Oats;
  - R: Ralston Purina;
- type: cold or hot.
- calories: calories per serving.
- protein: grams of protein.
- fat: grams of fat.
- sodium: milligrams of sodium.
- fiber: grams of dietary fiber.
- carbo: grams of complex carbohydrates.
- sugars: grams of sugars.
- potass: milligrams of potassium.
- vitamins: vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended.
- shelf: display shelf (1, 2, or 3, counting from the floor).
- weight: weight in ounces of one serving.
- cups: number of cups in one serving.



More information is available at: <https://community.amstat.org/stat-computing/data-expo/data-expo-1993>

**Source**

The dataset originates from the 1993 ASA Statistical Graphics Exposition, organized by the American Statistical Association. It is publicly available from DASL: <http://lib.stat.cmu.edu/datasets/1993.expo/cereal>

---

## churn Churn dataset for Credit Card Customers

---

### Description

Customer *churn* occurs when customers stop doing business with a company, also known as customer attrition. The dataset contains 10,127 rows (customers) and 21 columns (features). The “churn” column is our target which indicate whether customer churned (left the company) or not.

### Usage

```
df=datadepot.load("churn")
```

### Format

The churn dataset contains 10,127 rows (customers) and 21 columns.

The 21 variables are:

- `customer_id`: Customer ID.
- `gender`: Whether the customer is a male or a female.
- `age`: Customer's Age in Years.
- `educaton`: Educational Qualification of the account holder (example: high school, college graduate, etc.)
- `marital_status`: Married, Single, Divorced, Unknown
- `income`: Annual Income (in Dollar). Category of the account holder (< \$40K, \$40K - 60K, \$60K - \$80K, \$80K-\$120K, > \$120K).
- `dependent_counts`: Number of dependent counts.
- `card_category`: Type of Card (Blue, Silver, Gold, Platinum).
- `months_on_book`: Period of relationship with bank.
- `relationship_count`: Total number of products held by the customer.
- `months_inactive`: Number of months inactive in the last 12 months.
- `contacts_count_12`: Number of Contacts in the last 12 months.
- `credit_limit`: Credit Limit on the Credit Card.
- `revolving_balance`: Total Revolving Balance on the Credit Card.
- `open_to_buy`: Open to Buy Credit Line (Average of last 12 months).
- `transaction_amount_Q4_Q1`: Change in Transaction Amount (Q4 over Q1).
- `transaction_amount_12`: Total Transaction Amount (Last 12 months).
- `transaction_count`: Total Transaction Count (Last 12 months).
- `transaction_change`: Change in Transaction Count (Q4 over Q1).
- `utilization_ratio`: Average Card Utilization Ratio.
- `churn`: Whether the customer churned or not (yes or no).

**Source**

This dataset is publicly available from Kaggle: <https://www.kaggle.com/sakshigoyal7/credit-card-customers>

---

## churn\_ibm Telco Customer Churn dataset

---

### Description

Customer *churn* occurs when customers stop doing business with a company, also known as customer attrition. This synthetic dataset from IBM contains 7043 rows (customers) and 21 columns (features). The “churn” column is our target which indicate whether customer churned (left the company) or not.

### Usage

```
df=datadepot.load(“churn_ibm”)
```

### Format

The churn\_ibm dataset contains 7,043 rows (customers) and 21 columns.

The 21 variables are:

- customer\_id: Customer ID.
- gender: Whether the customer is a male or a female.
- senior\_citizen: Whether the customer is a senior citizen or not (1, 0).
- partner: Whether the customer has a partner or not (yes, no).
- dependent: Whether the customer has dependents or not (yes, no).
- tenure: Number of months the customer has stayed with the company.
- phone\_service: Whether the customer has a phone service or not (yes, no).
- multiple\_lines: Whether the customer has multiple lines or not (yes, no, no phone service).
- internet\_service: Customer’s internet service provider (DSL, fiber optic, no).
- online\_security: Whether the customer has online security or not (yes, no, no internet service).
- online\_backup: Whether the customer has online backup or not (yes, no, no internet service).
- device\_protection: Whether the customer has device protection or not (yes, no, no internet service).
- tech\_support: Whether the customer has tech support or not (yes, no, no internet service).
- streaming\_tv: Whether the customer has streaming TV or not (yes, no, no internet service).
- streaming\_movie: Whether the customer has streaming movies or not (yes, no, no internet service).
- contract: the contract term of the customer (month to month, 1 year, 2 year).
- paperless\_bill: Whether the customer has paperless billing or not (yes, no).

- `payment_method`: the customer's payment method (electronic check, mail check, bank transfer, credit card).
- `monthly_charge`: the amount charged to the customer monthly.
- `total_charges`: the total amount charged to the customer.
- `churn`: Whether the customer churned or not (yes or no).

**Source**

This dataset is publicly available from Github: <https://github.com/IBM/telco-customer-churn-on-icp4d>.

**References**

<https://www.ibm.com/docs/en/cognos-analytics/12.1.0?topic=samples-telco-customer-churn>

---

## churn\_mlc Churn dataset

---

### Description

This synthetic dataset from MLC++ machine learning software is used for modeling customer churn. Customer churn occurs when customers stop doing business with a company, also known as customer attrition. The dataset contains 5,000 rows (customers) and 20 columns (features). The **churn** column is our target which indicate whether customer churned (left the company) or not.

### Usage

```
df=datadepot.load("churn_mlc")
```

### Format

The churn\_mlc dataset contains 5,000 rows (customers) and 20 columns.

The 20 variables are:

- state: Categorical, for the 51 states and the District of Columbia.
- account\_length: Count, how long account has been active.
- area\_code: Categorical.
- international\_plan: Categorical (1/0), whether the customer has an international plan.
- voice\_mail\_plan: Categorical (1/0), whether the customer has a voice mail plan.
- number\_vmail\_messages: Count, number of voice mail messages.
- total\_day\_minutes: Continuous, minutes customer used service during the day.
- total\_day\_calls: Count, total number of calls during the day.
- total\_day\_charge: Continuous, total charge during the day.
- total\_eve\_minutes: Continuous, minutes customer used service during the evening.
- total\_eve\_calls: Count, total number of calls during the evening.
- total\_eve\_charge: Continuous, total charge during the evening.
- total\_night\_minutes: Continuous, minutes customer used service during the night.
- total\_night\_calls: Count, total number of calls during the night.
- total\_night\_charge: Continuous, total charge during the night.
- total\_intl\_minutes: Continuous, minutes customer used service to make international calls.
- total\_intl\_calls: Count, total number of international calls.
- total\_intl\_charge: Continuous, total international charge.
- number\_customer\_service\_calls: Count, number of calls to customer service.
- churned: Categorical (Yes/No), whether the customer has left the company.

## Source

This dataset was originally provided by MLC++. The original MLC++ site is no longer available, but the data can be found here:

- OpenML: <https://openml.org/d/46915>
- data.world: <https://data.world/earino/churn>
- modeldata package (available on CRAN): <https://cran.r-project.org/web/packages/modeldata/index.html>

## References

- Marcoulides, G. A. (2005). Discovering Knowledge in Data: an Introduction to Data Mining. Journal of the American Statistical Association, 100(472), 1465. <https://doi.org/10.1198/jasa.2005.s61>
- Saha, S., Saha, C., Haque, M. M., Alam, M. G. R., & Talukder, A. (2024). ChurnNet: Deep learning enhanced customer churn prediction in telecommunication industry. IEEE access, 12, 4471-4484.

---

## **covid** COVID-19 Coronavirus dataset

---

### **Description**

COVID-19 Coronavirus data - daily (up to 14 December 2020).

### **Usage**

```
df=datadepot.load("covid")
```

### **Format**

The covid dataset contains 61,900 rows and 12 columns.

More information is available at: <https://data.europa.eu/data/datasets/covid-19-coronavirus-data-daily-up-to-14-december-2020>

### **Source**

This dataset is publicly available from the European Union's data repository: <https://data.europa.eu/data/datasets/covid-19-coronavirus-data-daily-up-to-14-december-2020>



---

## **credit** South German Credit dataset

---

### **Description**

This dataset classifies people described by a set of attributes as good or bad credit risks.

### **Usage**

```
df=datadepot.load("credit")
```

### **Format**

The dataset contains 1,000 rows and 20 columns (variables).

The 20 variables are:

- status
- duration
- credit\_history
- purpose
- amount
- savings
- employment\_duration
- installment\_rate
- personal\_status\_sex
- other\_debtors
- present\_residence
- property
- age
- other\_installment\_plans
- housing
- number\_credits
- job
- people\_liable
- telephone
- foreign\_worker
- credit\_risk

### **Source**

The dataset is publicly available from the UCI machine learning repository: <https://doi.org/10.24432/C5QG88>

---

## **credit\_card** Credit Card Fraud dataset

---

### **Description**

Credit Card Transactions

### **Usage**

```
df=datadepot.load("credit_card")
```

### **Format**

The dataset contains 284,807 rows and 31 columns.

The dataset contains the following columns:

- class: Indicator of Fraud (1) or Non-Fraud (0)
- time: Seconds elapsed between each transaction and the first transaction in the dataset.
- amount: Transaction amount
- features v1, v2, ... v28: Transformed features, obtained through Principal Component Analysis

More information is available at: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

### **Source**

This dataset is publicly available from Kaggle: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

---

## cpu CPU dataset

---

### Description

This dataset contains detailed specifications and performance metrics for a range of computer processors (CPUs). It includes hardware characteristics such as core and thread counts, clock speeds, cache size, and thermal design power (TDP), along with market price information. The dataset is designed to analyze price-to-performance trade-offs.

### Usage

```
df=datadepot.load("cpu")
```

### Format

The dataset contains 45 rows and 12 columns.

- brand: The brand of the CPU (AMD or Intel)
- model: The model of the processor
- architecture: The microarchitecture or generation family of the CPU
- p\_cores: Number of performance cores (P-cores)
- e\_cores: Number of efficiency cores (E-cores)
- threads: Number of logical threads the CPU can execute simultaneously
- base\_ghz: The base operating frequency of the CPU in gigahertz
- boost\_ghz: The maximum turbo/boost frequency the CPU in gigahertz
- cache: Total L3 cache size in MB
- tdp: Typical thermal design power (TDP) in watts under standard load conditions
- price: Market price of the CPU (in US Dollars)

### Source

Hardware specifications are based on publicly available manufacturer data. Price data was collected via Google search during Spring 2026 and reflects approximate retail market prices at that time.

### Creator

J. van Raak

---

## **diamonds** Diamonds dataset

---

### **Description**

The diamonds dataset from ggplot2 is a comprehensive collection of data about diamonds, widely used in data science to practice visualization, statistical modeling, and machine learning. It includes detailed characteristics of individual diamonds and their corresponding prices. The dataset contains detailed information on over 50,000 diamonds, including both physical characteristics and pricing.

### **Usage**

```
df=datadepot.load("diamonds")
```

### **Format**

The diamonds dataset contains 53,940 rows and 10 columns.

The 10 variables are:

- carat: weight of the diamond (ranging from 0.2 to 5.01),
- cut: quality of the cut (Fair, Good, Very Good, Premium, Ideal),
- color: color grade, from D (most colorless) to J (least colorless),
- clarity: clarity grade, from I1 (least clear) to IF (flawless),
- depth: total depth percentage calculated as:  $2 * z / (x + y)$ ,
- table: width of the top facet relative to the widest point (43-95%),
- x: length in mm
- y: width in mm
- z: depth in mm
- price: price in US dollars (ranging from \$326 to \$18,823).

### **Source**

This dataset is publicly available in the ggplot2 R package: <https://ggplot2.tidyverse.org/reference/diamonds.html>

For more information related to the dataset see:

<https://search.r-project.org/CRAN/refmans/nodbi/html/diamonds.html>

---

## **drug** Drug Classification dataset

---

### **Description**

A synthetically generated dataset of 200 patients that includes their age, sodium-to-potassium (Na/K) ratio, and the prescribed drug type.

### **Usage**

```
df=datadepot.load("drug")
```

### **Format**

The drug dataset contains 200 rows and 3 columns.

The 3 variables are:

- age: patient age,
- ratio: sodium-to-potassium (Na/K) ratio,
- type: prescribed drug type with three levels: A, B and C.

### **Source**

This dataset is generated for the book ‘Data Science Foundations and Machine Learning’.

### **Creator**

Reza Mohammadi

---

## gapminder Gapminder dataset

---

### Description

The dataset covers a wide range of countries across all continents and allows for longitudinal analysis of economic growth, demographic changes, and improvements in health over nearly seven decades. It is suitable for research, visualization, and modeling related to development, public policy, and global trends.

### Usage

```
df=datadepot.load("gapminder")
```

### Format

This dataset provides comprehensive historical data on key socio-economic indicators for countries around the world, spanning the years 1950 through 2019. It contains 13650 observations. The dataset includes the following variables:

- country: Name of the country.
- continent: Continent to which the country belongs.
- year: The calendar year of the observation.
- gdp: The gross domestic product per person, measured in constant international dollars (adjusted for inflation and purchasing power parity).
- population: Total population of the country.
- life\_expectancy: Average life expectancy at birth, in years.
- iso\_alpha: Three-letter standardized country code.

### Source

Free data sourced from the World Bank via Gapminder.org. Available under the Creative Commons Attribution (CC BY) license. This data is publicly available from Gapminder: <https://www.gapminder.org/data/>

---

## hotel\_city Hotel Cancellations: City Hotels

---

### Description

The dataset from Antonio, Almeida and Nunes (2019) examines hotel cancellations for city hotels.

### Usage

```
df=datadepot.load("hotel_city")
```

### Format

The dataset contains 79,330 observations and 31 columns:

- `is_canceled`: Value indicating if the booking was canceled (1) or not (0)
- `lead_time`: Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
- `arrival_date_year`: Year of arrival date
- `arrival_date_month`: Month of arrival date
- `arrival_date_week_number`: Week number of year for arrival date
- `arrival_date_day_of_month`: Day of arrival date
- `stays_in_weekend_nights`: Number of weekend nights the guest stayed or booked to stay at the hotel
- `stays_in_week_nights`: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
- `adults`: Number of adults
- `children`: Number of children
- `babies`: Number of babies
- `meal`: Type of meal booked. Categories are presented in standard hospitality meal packages:
  - Undefined
  - SC – no meal package
  - BB – Bed & Breakfast
  - HB – Half board (breakfast and one other meal – usually dinner)
  - FB – Full board (breakfast, lunch and dinner)
- `country`: Country of origin
- `market_segment`: Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- `distribution_channel`: Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”

- `is_repeated_guest`: Value indicating if the booking name was from a repeated guest (1) or not (0)
- `previous_cancellations`: Number of previous bookings that were cancelled by the customer prior to the current booking
- `previous_bookings_not_canceled`: Number of previous bookings not cancelled by the customer prior to the current booking
- `reserved_room_type`: Code of room type reserved. Code is presented instead of designation for anonymity reasons
- `assigned_room_type`: Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
- `booking_changes`: Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
- `deposit_type`: Indication of deposit status for the booking. Categories include:
  - No Deposit – no deposit was made
  - Non Refund – a deposit was made in the value of the total stay cost
  - Refundable – a deposit was made with a value under the total cost of stay
  - Canceled – booking was canceled by the customer
  - Check-Out – customer has checked in but already departed
  - No-Show – customer did not check-in and did inform the hotel of the reason why
- `agent`: ID of the travel agency that made the booking
- `company`: ID of the company/entity responsible for the booking (anonymized)
- `days_in_waiting_list`: Number of days the booking was in the waiting list before it was confirmed to the customer
- `customer_type`: Type of booking, assuming one of four categories:
  - Contract – booking has an allotment or contract
  - Group – booking is associated to a group
  - Transient – not part of a group or contract
  - Transient-party – transient booking associated with other transient bookings
- `adr`: Average Daily Rate (sum of all lodging transactions divided by total staying nights)
- `required_car_parking_spaces`: Number of car parking spaces required by the customer
- `total_of_special_requests`: Number of special requests made by the customer (e.g. twin bed, high floor)
- `reservation_status`: Reservation last status (e.g. Canceled, Check-Out, No-Show)
- `reservation_status_date`: Date at which the last status was set

## Source

This dataset is publicly available from: <https://doi.org/10.1016/j.dib.2018.11.126>

## References



Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. *Data in brief*, 22, 41-49.

---

## **hotel\_resort** Hotel Cancellations: Resort Hotels

---

### **Description**

The dataset from Antonio, Almeida and Nunes (2019) examines hotel cancellations for resort hotels.

### **Usage**

```
df=datadepot.load("hotel_resort")
```

### **Format**

The dataset contains 40,060 observations and 31 columns:

- `is_canceled`: Value indicating if the booking was canceled (1) or not (0)
- `lead_time`: Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
- `arrival_date_year`: Year of arrival date
- `arrival_date_month`: Month of arrival date
- `arrival_date_week_number`: Week number of year for arrival date
- `arrival_date_day_of_month`: Day of arrival date
- `stays_in_weekend_nights`: Number of weekend nights the guest stayed or booked to stay at the hotel
- `stays_in_week_nights`: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
- `adults`: Number of adults
- `children`: Number of children
- `babies`: Number of babies
- `meal`: Type of meal booked. Categories are presented in standard hospitality meal packages:
  - Undefined
  - SC – no meal package
  - BB – Bed & Breakfast
  - HB – Half board (breakfast and one other meal – usually dinner)
  - FB – Full board (breakfast, lunch and dinner)
- `country`: Country of origin
- `market_segment`: Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- `distribution_channel`: Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”

- `is_repeated_guest`: Value indicating if the booking name was from a repeated guest (1) or not (0)
- `previous_cancellations`: Number of previous bookings that were cancelled by the customer prior to the current booking
- `previous_bookings_not_canceled`: Number of previous bookings not cancelled by the customer prior to the current booking
- `reserved_room_type`: Code of room type reserved. Code is presented instead of designation for anonymity reasons
- `assigned_room_type`: Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
- `booking_changes`: Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
- `deposit_type`: Indication of deposit status for the booking. Categories include:
  - No Deposit – no deposit was made
  - Non Refund – a deposit was made in the value of the total stay cost
  - Refundable – a deposit was made with a value under the total cost of stay
  - Canceled – booking was canceled by the customer
  - Check-Out – customer has checked in but already departed
  - No-Show – customer did not check-in and did inform the hotel of the reason why
- `agent`: ID of the travel agency that made the booking
- `company`: ID of the company/entity responsible for the booking (anonymized)
- `days_in_waiting_list`: Number of days the booking was in the waiting list before it was confirmed to the customer
- `customer_type`: Type of booking, assuming one of four categories:
  - Contract – booking has an allotment or contract
  - Group – booking is associated to a group
  - Transient – not part of a group or contract
  - Transient-party – transient booking associated with other transient bookings
- `adr`: Average Daily Rate (sum of all lodging transactions divided by total staying nights)
- `required_car_parking_spaces`: Number of car parking spaces required by the customer
- `total_of_special_requests`: Number of special requests made by the customer (e.g. twin bed, high floor)
- `reservation_status`: Reservation last status (e.g. Canceled, Check-Out, No-Show)
- `reservation_status_date`: Date at which the last status was set

## Source

This dataset is publicly available from: <https://doi.org/10.1016/j.dib.2018.11.126>

## References

Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. *Data in brief*, 22, 41-49.

---

## house House dataset

---

### Description

The house dataset contains 6 features and 414 records. The target feature is *unit\_price* and the remaining 5 variables are predictors.

### Usage

```
df=datadepot.load("house")
```

### Format

The house dataset contains 414 rows and 6 columns. The 6 variables are:

- house\_age: house age (numeric, in year).
- distance\_to\_mrt: distance to the nearest MRT station (numeric).
- stores\_number: number of convenience stores (numeric).
- latitude: latitude (numeric).
- longitude: longitude (numeric).
- unit\_price: house price of unit area (numeric).

### Source

The data is publicly available from UCI Irvine: <https://archive.ics.uci.edu/dataset/477/real+estate+valuation+data+set>

---

## **house\_price** House Price dataset

---

### **Description**

This dataset, created by Dean De Cock, contains 1460 rows and 81 columns (features). The **saleprice** column is the target.

### **Usage**

```
df=datadepot.load("house_price")
```

### **Format**

The house\_price dataset contains 1460 rows and 81 columns. More information about the variables can be found at: <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data>

### **Source**

The data is publicly available from Kaggle: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

---

## insurance Insurance dataset

---

### Description

The insurance dataset contains 7 columns and 1,338 records. The target feature is **charge** and the remaining 6 variables are predictors.

### Usage

```
df=datadepot.load("insurance")
```

### Format

The synthetic insurance dataset contains 1,338 rows (customers) and 7 columns.

The 7 variables are:

- age: age of primary beneficiary.
- bmi: body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight ( $\text{kg} / \text{m}^2$ ) using the ratio of height to weight, ideally 18.5 to 24.9.
- children: Number of children covered by health insurance / Number of dependents.
- smoker: Smoking as a factor with 2 levels, yes, no.
- gender: insurance contractor gender, female, male.
- region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- charge: individual medical costs billed by health insurance.

A detailed description of the dataset can be found at: <https://www.kaggle.com/mirichoi0218/insurance>

### Source

The dataset is publicly available from Kaggle: <https://www.kaggle.com/datasets/mirichoi0218/insurance/dat>

### Reference

Brett Lantz (2019). Machine Learning with R: Expert techniques for predictive modeling. *Packt Publishing Ltd*.

---

## **las\_vegas** Las Vegas Strip

---

### **Description**

The dataset includes both quantitative and qualitative features extracted from TripAdvisor reviews of 21 hotels located on the Las Vegas Strip.

### **Usage**

```
df=datadepot.load("las_vegas")
```

### **Format**

The dataset contains 504 observations and 20 columns:

- user\_\_country
- nr\_\_reviews
- nr\_\_hotel\_\_reviews
- helpful\_\_votes
- score
- period\_of\_\_stay
- traveler\_\_type
- pool
- gym
- tennis\_\_court
- spa
- casino
- free\_\_internet
- hotel\_\_name
- hotel\_\_stars
- nr\_\_rooms
- user\_\_continent
- member\_\_years
- review\_\_month
- review\_\_weekday

### **Source**

This dataset is publicly available from UC Irvine Machine Learning Repository: <https://doi.org/10.24432/C5QG7W>

### **Creators**

S. Moro, P. Rita and J. Coelho



**Remarks**

Replaced spaces and dots in column names by underscores.

---

## **loan** Loan Approval

---

### **Description**

The loan approval dataset is a collection of financial records and associated information used to determine the eligibility of individuals or organizations for obtaining loans from a lending institution.

### **Usage**

```
df=datadepot.load("loan")
```

### **Format**

The dataset contains 4,269 observations and 13 columns:

- loan\_id
- no\_of\_dependents: Number of Dependents of the Applicant
- education: Education of the Applicant
- self\_employed: Employment Status of the Applicant
- income\_annum: Annual Income of the Applicant
- loan\_amount: Loan Amount
- loan\_term: Loan Term in Years
- cibil\_score
- Credit Score
- residential\_assets\_value
- loan\_status

### **Source**

This dataset is publicly available from Kaggle: <https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data>

---

## **machine\_failure** Machine Failure dataset

---

### **Description**

This dataset contains sensor data collected from various machines, with the aim of predicting machine failures in advance. It includes a variety of sensor readings as well as the recorded machine failures.

### **Usage**

```
df=datadepot.load("machine_failure")
```

### **Format**

The dataset contains 944 observations and 10 columns:

- footfall: The number of people or objects passing by the machine.
- tempmode: The temperature mode or setting of the machine.
- aq: Air quality index near the machine.
- uss: Ultrasonic sensor data, indicating proximity measurements.
- cs: Current sensor readings, indicating the electrical current usage of the machine.
- voc: Volatile organic compounds level detected near the machine.
- rp: Rotational position or RPM (revolutions per minute) of the machine parts.
- ip: Input pressure to the machine.
- temperature: The operating temperature of the machine.
- fail: Binary indicator of machine failure (1 for failure, 0 for no failure).

### **Source**

This dataset is publicly available from Kaggle: <https://www.kaggle.com/datasets/umerrtx/machine-failure-prediction-using-sensor-data>

---

## mpg Auto MPG dataset

---

### Description

The Auto MPG dataset contains information on various car models from the 1970s and 1980s, with the goal of predicting fuel efficiency (miles per gallon, `mpg`). It includes attributes describing engine characteristics, vehicle weight, performance, model year, origin, and car name.

### Usage

```
df=datadepot.load("mpg")
```

### Format

The Auto MPG dataset contains 398 observations (cars) and 9 columns:

- `mpg`: miles per gallon, a continuous variable measuring fuel efficiency.
- `cylinders`: number of cylinders in the engine, a discrete factor with typical values 3, 4, 5, 6, 8.
- `displacement`: engine displacement in cubic inches, a continuous variable representing engine size.
- `horsepower`: engine power in horsepower, a continuous variable (may have missing values).
- `weight`: vehicle weight in pounds, a continuous variable.
- `acceleration`: time to accelerate from 0 to 60 mph in seconds, a continuous variable.
- `model_year`: year of the car model, a discrete variable usually coded as two digits (e.g., 70 = 1970).
- `origin`: origin of the car, a factor with 3 levels (1 = USA, 2 = Europe, 3 = Japan).
- `car_name`: car model name, a string variable for identification.

### Source

This dataset is publicly available from the UCI machine learning repository: <https://archive.ics.uci.edu/dataset/9/auto+mpg>

---

## **red\_wines** Red Wines dataset

---

### **Description**

The red\_wines datasets are related to red variants of the Portuguese “Vinho Verde” wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.). The dataset can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

### **Usage**

```
df=datadepot.load("red_wines")
```

### **Format**

The red\_wines dataset contains 1599 rows and 12 columns.

The 12 variables are:

Input variables (based on physicochemical tests):

- fixed\_acidity
- volatile\_acidity
- citric\_acid
- residual\_sugar
- chlorides
- free\_sulfur\_dioxide
- total\_sulfur\_dioxide
- density
- ph
- sulphates
- alcohol

Output variable (based on sensory data)

- quality: score between 0 and 10.

**Source**

The dataset is publicly available from the UCI machine learning repository: <https://archive.ics.uci.edu/dataset/186/wine+quality>

**Reference**

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4), 547-553.

---

## **vehicle** Vehicle Prices

---

### **Description**

This dataset contains information about used cars and resale values. The information is obtained from CarDekho.

### **Usage**

```
df=datadepot.load("vehicle")
```

### **Format**

The vehicle dataset contains 4,340 rows and 8 columns.

The variables are:

- name: brand and model of the car
- year: year the car was manufactured
- selling\_price: listed selling price of the car (in Indian Rupees)
- km\_driven: total distance driven (in kilometers)
- fuel: type of fuel used by the car
- seller\_type: type of seller (Individual or Dealer)
- transmission: transmission type (Manual or Automatic)
- owner: Ownership statues (e.g., first owner, second owner, etc.)

### **Source**

The dataset is publicly available from the Kaggle: <https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho>

### **Creators**

Nehal Birla, Nishant Verma and Nikhil Kushwaha

---

## **white\_wines** White Wines dataset

---

### **Description**

The white\_wines datasets are related to white variants of the Portuguese “Vinho Verde” wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.). The dataset can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

### **Usage**

```
df=datadepot.load("white_wines")
```

### **Format**

The white\_wines dataset contains 4898 rows and 12 columns.

Input variables (based on physicochemical tests):

- fixed\_acidity
- volatile\_acidity
- citric\_acid
- residual\_sugar
- chlorides
- free\_sulfur\_dioxide
- total\_sulfur\_dioxide
- density
- ph
- sulphates
- alcohol

Output variable (based on sensory data)

- quality: score between 0 and 10.

### **Source**

The dataset is publicly available from the UCI machine learning repository: <https://archive.ics.uci.edu/dataset/186/wine+quality>

### **References**



Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4), 547-553.

---

## **wholesale** Wholesale dataset

---

### **Description**

The dataset contains clients of a wholesale distributor and their annual spending (in monetary units) across different product categories.

### **Usage**

```
df=datadepot.load("wholesale")
```

### **Format**

The wholesale dataset contains 440 rows and 7 columns.

The 8 variables are:

channel: customer channel (Horeca: hotel/restaurant/café or retail) (nominal) region: customer region (Lisbon, Oporto, or other) (nominal) fresh: annual spending (m.u.) on fresh products (continuous) milk: annual spending (m.u.) on milk products (continuous) grocery: annual spending (m.u.) on grocery products (continuous) frozen: annual spending (m.u.) on frozen products (continuous) detergents\_paper: annual spending (m.u.) on detergents and paper products (continuous) delicatessen: annual spending (m.u.) on delicatessen products (continuous)

### **Source**

The dataset is publicly available from the UCI machine learning repository: <https://archive.ics.uci.edu/dataset/292/wholesale+customers>

### **Creator**

Margarida Cardoso