Hyperiondev

TASK

Data Analysis - Preprocessing

Visit our website

Introduction

WELCOME TO THE DATA ANALYSIS - PREPROCESSING TASK!

In this task, we are going to work on how to handle categorical and missing data as well as scaling data. Preprocessing data allows us to gain insights into the data and highlights problems such as missing or corrupted data. It is also important to transform data to build applicable machine learning models.

WORKING WITH CATEGORICAL DATA

As discussed previously, categorical data deals with discrete (individually separate and distinct) data that fits neatly into a number of categories. Data tables and visualisations are often used to analyse this type of data.

Data tables are often used to count the number of variables within a particular category. For example, you may want to analyse a dataset that stores data about HyperionDev students. The dataset could contain categorical data such as which Bootcamp a student is currently registered for. Therefore, it could be useful to create a data table that displays the total number of students registered for each Bootcamp to help analyse this dataset.

Two-way tables are useful when analysing how data in two categorical variables relate. Consider this example by Lacey: "suppose a survey was conducted of a group of 20 individuals, who were asked to identify their hair and eye colour.
A two-way table presenting the results might appear as follows":

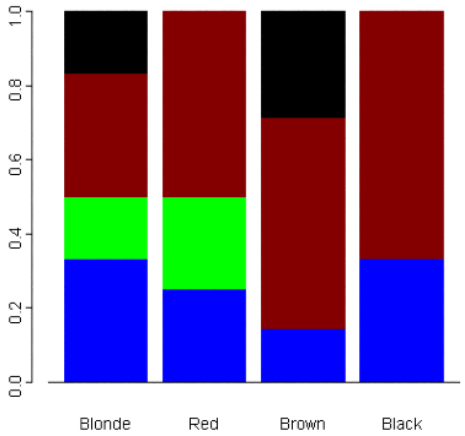
Hair Colour	**Eye Colour**				
	Blue	*Green*	*Brown*	*Black*	**Total**
Blonde	2	1	2	1	**6**
Red	1	1	2	0	**4**
Brown	1	0	4	2	**7**
Black	1	0	2	0	**3**
Total	**5**	**2**	**10**	**3**	**20**

(Lacey, n.d)

The totals for each category are known as marginal distributions. Two-way tables

are often converted to percentages to make this data easier to analyse. For example, it may be more helpful to know what percentage of people surveyed have blue eyes or what percentage of people with red hair have green eyes.

Segmented bar graphs are also often useful for analysing categorical data. For example, notice how the segmented bar graph below can be used to represent the information about people's eye colour (colour-coded) and hair colour:



(Lacey, n.d)

Sometimes it is helpful to introduce a categorical variable into a dataset to make continuous variables easier to analyse. For example, suppose you had a dataset that stores the weight for several boxers (continuous variable). In that case, it may be useful to introduce a weight range variable (e.g. minimum_weight, light_fly_weight, fly_weight, etc.).

MISSING VARIABLES

As you have already learned before you can do any useful analysis, it is important to clean your dataset. One problem that you will often encounter when cleaning data is missing data. Data can have missing values for many reasons. For example, an observation (a piece of data) may not have been collected or recorded, or data corruption may have occurred. Data corruption is when data becomes unusable, unreadable or in some other way inaccessible to a user or application. There is no perfect way of dealing with missing data! The approach you use will depend on a good understanding of the data and the effect that the method you use will have on the dataset. The common approaches to handling missing data are trying to find missing data, removing observations with missing data, or using an imputation method to replace missing data with a substitute value. However, before deciding how to deal with missing data, it is essential to understand why it is missing.

Missing data can be categorised into three types:

- **Missing Completely at Random (MCAR)**: This means that the missing data is **not in any way related** to any other data in the dataset.
    - For example, data may be missing because a test paper was lost or a blood sample was damaged. This means there is no way to predict what was missing (and therefore no reasonable way to guess what those values are). This is entirely random.
- **Missing at Random (MAR)**: This is when the missing data is somehow related to other data in the dataset. If data is MAR, it can sometimes be predicted based on other data in the dataset. In other words, MAR data **can be explained by other data measured in the dataset**. This is due to implicit biases in the data itself. These biases may be used partially to gain a more realistic statistic.
    - For example, consider a survey on depression. Studies have shown that males are less likely to fill in surveys about depression severity. In other words, this missingness can be determined by their gender (which was noted in the dataset).
- **Missing Not at Random (MNAR)**: This occurs when there is a direct relation to some missing data that the researchers haven't measured. Like with MAR, certain biases affect the values in the dataset. However, these biases are from **factors not measured in the dataset.**
    - An example of this is in COVID reporting. Since restrictions were lifted and it had a lower impact on our lives, we saw a drop in the reported COVID cases. This is not because lifting restrictions makes the disease go away - it is just that people are less likely to get tested! This is something that scientists cannot measure and, therefore, can't be imputed.

IMPUTATION METHODS

In some cases, missing data can be replaced with substitute values instead of removing entries with missing values. Here are some methods that can be used to calculate what those substitute values should be.

Mean, median and mode imputation

Using the measures of central tendency involves substituting the missing values with the mean or median for numerical variables and the mode for categorical

variables. This imputation technique works well when the values are missing completely at random (MCAR). One disadvantage is that mean imputation reduces variance in the dataset.

Imputation with linear regression

This imputation technique utilises variables from the observed data to replace the missing values with predicted values from a regression model. Complete observations are used to generate the regression equation; the equation is then used to predict missing values for incomplete observations. In an iterative process, values for the missing variable are inserted, and then all cases are used to predict the dependent variable. These steps are repeated until there is little difference between the predicted values from one step to the next; that is, they converge. The major drawback of using this method is that it reduces variability. Though we have yet to introduce regression, it is important to keep this in mind.

K-Nearest Neighbour (KNN) imputation

For K-nearest neighbour imputation, the values are obtained by using a similarity-based method that relies on distance metrics (Euclidean distance, Jaccard similarity, Minkowski norm etc.). This method can predict both discrete and continuous attributes. KNN works by finding other observations that are most similar to the observation with the missing value. For example, if the observation is 'Female' and 'Asian', we can find other users similar to her and get the mean or mode of the missing value from the other observations. The main disadvantage of using KNN imputation is that it becomes time-consuming when analysing large datasets because it searches for similar instances in the entire dataset.

Whether we remove observations with missing data or substitute the missing value with another value, we must be careful not to create bias! Rumsey defines **statistical bias** as the “systematic favouritism that is present in the data collection process, resulting in lopsided, misleading results.” When we remove or add substituted data, we could cause bias by creating a dataset favouring a certain idea. For example, if you remove mainly data about people with low income or assume that all those with a missing income value are high income, your findings could be distorted.

Extra resource

Explore the **Missing Data** additional reading in this task folder to learn more about missing data and other imputation methods to replace missing data with substitute values.

FEATURE SCALING: NORMALISATION AND STANDARDISATION

Another common problem we encounter when trying to analyse data is having different units of measurement for a particular variable. For example, if you wanted to compare housing prices in different countries, you would have different datasets that store price values with different currencies. Clearly, it is not easy to compare these values. To get rid of the unit of measurement, we scale data by either normalising or standardising the data. In this section, we briefly consider some methods of doing this.

Normalisation

Normalisation involves scaling a variable to have a value between 0 and 1. Normalisation is often done using the formula below:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This helps to eliminate units of measurement for data. For example, suppose you are working with a dataset containing variables containing different measurement units, such as kilometres, miles, etc. In that case, using scaling reduces all the variables to a scale between 0 and 1, removing the need for units of measurement. This helps by allowing you to compare data from different sources. For example, if you wanted to compare housing prices in America vs Germany, you would have to change your data to measure the prices using the same scale.

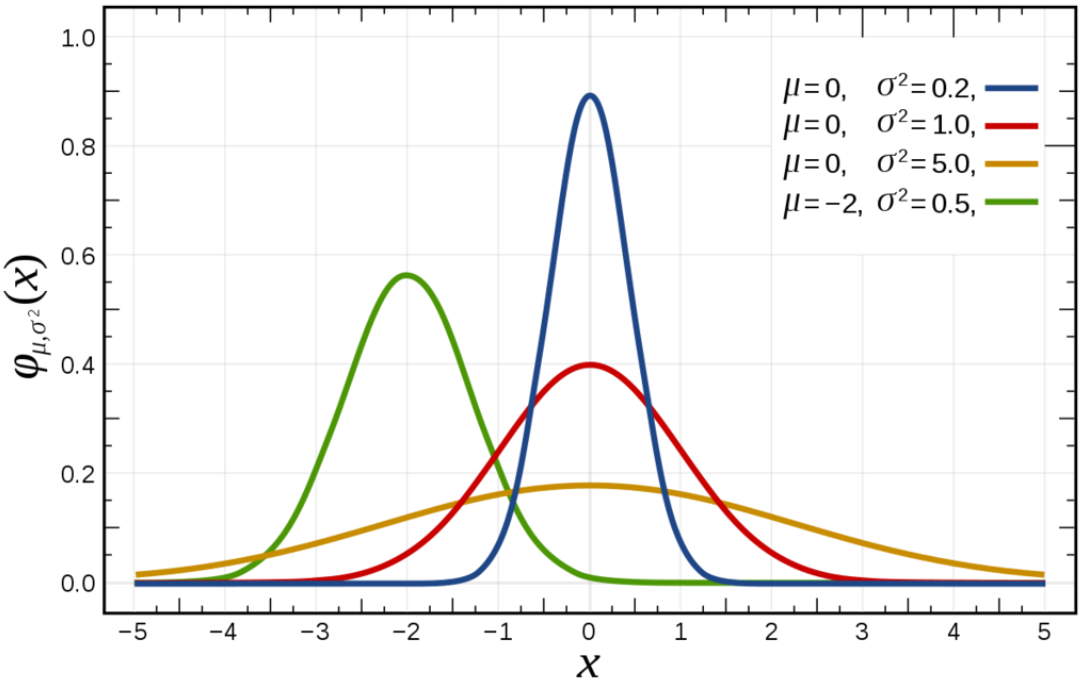
Standardisation

Standardisation involves scaling the data to have a mean (average) of 0 and a standard deviation (the average deviation from the mean in distribution) of 1. Standardisation uses this formula:

$$z = \frac{x_i - \mu}{\sigma}$$

This is a better form of feature scaling than normalisation, as the data doesn't restrict the scale as much as for normalisation. Consequently, outliers won't affect the data range as much as normalisation. However, in order to standardise data, you must ensure that your data follows a **Gaussian distribution**.

Note a Gaussian distribution or normal distribution has a bell shape, as shown in the image below.



Source: **Wikimedia**

Normalisation vs. Standardisation

Now we know the difference between normalisation and standardisation. Normalisation restricts all values between 0 and 1, and standardisation causes the data to have a mean of 0 and a standard deviation of 1. But how do we know when to use which one? Quite simply, you must **standardise** the data when it follows a **Gaussian** distribution. If not, **normalise** the data.

Practical Task 1

Open the **data_preprocessing.ipynb** file and explore the examples, then follow these steps in the notebook:

1. Read in the **store_income_data_task.csv** file.
2. Display the first five observations.
3. Get the number of missing values per column and print the results.

4. Write a note on why you think we have missing data on the following three columns: **store_email**, **department**, and **country**.
    - Remember to classify them according to the three categories of missingness we have considered.

Practical Task 2

Follow these steps:

This task handles the normalisation and standardisation of variables. For more information about normalisation and standardisation, see **here**. Continue to task 2 in **data_preprocessing.ipynb**, in which you do the following:

1. For the following examples, decide whether normalisation or standardisation makes more sense:
    a. You want to build a linear regression model to predict someone's grades, given how much time they have spent on various activities during a typical school week. You notice that your measurements for how much time students spend studying aren't normally distributed: some students spend almost no time studying, while others study for four or more hours daily. Should you normalise or standardise this variable?
    b. You're still working with your student's grades, but you also want to include information on how students perform on several fitness tests. You have information on how many jumping jacks and push-ups each student can complete in a minute. However, you notice that students perform far more jumping jacks than push-ups: the average for the former is 40, and for the latter, only 10. Should you normalise or standardise this variable?

2. Visualise the "EG.ELC.ACCS.ZS" column from the countries dataset using a histogram. Then, scale the column using the appropriate scaling method (normalisation or standardisation). Finally, visualise the original and scaled data alongside each other. Note EG.ELC.ACCS.ZS is the percentage of the population with access to electricity.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved? Do you think we've done a good job?

Click here to share your thoughts anonymously.

REFERENCES

DeFilippi, R. R. (2018, April 29). Standardize or Normalize? Examples in Python. Retrieved April 13, 2019, from Medium.com: **https://medium.com/@rrfd/standardize-or-normalize-examples-in-python-e3f174b65dfc**

Kwak, S. K., & Kim, J. H. (2017). Statistical data preparation: management of missing values and outliers. Korean journal of anesthesiology, 70(4), 407–411. doi:10.4097/kjae.2017.70.4.407

Lacey, M. (n.d.). Categorical data. Retrieved April 30, 2019, from stats.yale.edu: **http://www.stat.yale.edu/Courses/1997-98/101/catdat.htm**

Laerd statistics. (n.d.). Types of variables. Retrieved April 30, 2019, from statistics.laerd.com: **https://statistics.laerd.com/statistical-guides/types-of-variable.php**

Rumsey, D. (2020). How to Identify Statistical Bias - dummies. Retrieved 25 August 2020, from **https://www.dummies.com/education/math/statistics/how-to-identify-statistical-bias/**

Swalin, A. (2018, January 31). How to Handle Missing Data. Retrieved May 13, 2019, from Towards Data Science: **https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4**