Lecture 8: data preprocessing

Introduction to Machine Learning

Sophie Robert

L3 MIASHS — Semestre 2

2023-2024

Introduction

Introduction

Some times, for optimum performance, the dataset needs to be *pre-processed* further than simply with visual aid and statistical estimators:

Introduction

Some times, for optimum performance, the dataset needs to be *pre-processed* further than simply with visual aid and statistical estimators:

- Dealing with missing data

Introduction

Some times, for optimum performance, the dataset needs to be *pre-processed* further than simply with visual aid and statistical estimators:

- Dealing with missing data
- Feature scaling

Introduction

Some times, for optimum performance, the dataset needs to be *pre-processed* further than simply with visual aid and statistical estimators:

- Dealing with missing data
- Feature scaling
- Outlier removal

In practice, most of *Data science* consists in cleaning up datasets.

Dealing with missing values

Introduction

During lab, we encountered the problem of missing values in a dataset.

Height	Weight
52	11
12	10
52	?

Question

Do you remember what were the suggested solutions ?

In this lecture, we are going to go a bit further.

Classification of missing values

Missing values can be classified into three categories (*Rubin*, 1976):

- **MCAR**: *Missing Completely at Random*
- **MAR**: *Missing At random*
- **MNAR**: *Missing Not At Random*

Data Missing Completely at Random

Missing Complety at Random

Data is said to be **Missing Completely at Random** (MCAR) if the probability of being missing is the same for each observation.

Data Missing Completely at Random

Missing Complelty at Random

Data is said to be **Missing Completely at Random** (MCAR) if the probability of being missing is the same for each observation.

Example: The measuring tool misfunctioned

Data Missing at Random

Missing at Random

Data is said to be **Missing at Random** (MAR) if the missing probability depends on some **observed** variables.

Data Missing at Random

Missing at Random

Data is said to be **Missing at Random** (MAR) if the missing probability depends on some **observed** variables.

Example:

- A participant in a poll is most likely to not question 2 if they did not answer question 1.
- Some participants do not have measures because of socio-ecoomic variables.

Data Missing Not at Random

Missing at Random

Data is said to be **Missing Not at Random** (MNAR) if the missing probability depends on some **unobserved** variables.

Data Missing Not at Random

Missing at Random

Data is said to be **Missing Not at Random** (MNAR) if the missing probability depends on some **unobserved** variables.

Example: A participant in a poll did not answer 1 because of their gender (which we do not know).

Dropping missing values

A simple solution can be to drop the records with the missing values (or the feature if too many missing values) but:

- May not have enough data to afford dropping it
- Missing values can bring information too

Dropping missing values

- **MCAR**: when dropping random values:
  - No bias
  - Reduce the quality of the model if dropping too much data

Dropping missing values

- **MCAR**: when dropping random values:
  - No bias
  - Reduce the quality of the model if dropping too much data
- **MAR**:
  - Removing missing values introduces bias
  - Missing values should be imputed

Dropping missing values

- **MCAR**: when dropping random values:
  - No bias
  - Reduce the quality of the model if dropping too much data
- **MAR**:
  - Removing missing values introduces bias
  - Missing values should be imputed
- **MNAR**:
  - Removing missing values introduces bias
  - Impute missing values is more difficult since we have no information about the generative process

Question

How can be handle missing data ?

Imputing missing values

Several methods are possible:

- With a unique value (mean, median, etc.)
- By the centroid* of the group (see in later lectures)
- Using k nearest neighbors

Using a unique value

Unique value imputation

Unique value imputation consists in giving a unique value to the missing values.
For quantitative variables: mean (not robust), median, mode . . .
For qualitative variables: separate category, most frequent class . . .

Using a unique value

Unique value imputation

Unique value imputation consists in giving a unique value to the missing values.
For quantitative variables: mean (not robust), median, mode ...
For qualitative variables: separate category, most frequent class ...

Advantages:

- Easy to understand
- Easy to compute

Using a unique value

Unique value imputation

Unique value imputation consists in giving a unique value to the missing values.
For quantitative variables: mean (not robust), median, mode . . .
For qualitative variables: separate category, most frequent class . . .

Advantages:

- Easy to understand
- Easy to compute

Limits:

- If many missing value, feature becomes unusable
- Not very suitable for MAR

Using k-nearest neighbors

Imputing missing values using KNN

KNN imputation consists in imputing the missing feature is imputed using values from the k nearest neighbors that have a value for the feature.

Using k-nearest neighbors

Imputing missing values using KNN

KNN imputation consists in imputing the missing feature is imputed using values from the k nearest neighbors that have a value for the feature.

Advantages:

- Takes into account dependence between variables

Using k-nearest neighbors

Imputing missing values using KNN

KNN imputation consists in imputing the missing feature is imputed using values from the k nearest neighbors that have a value for the feature.

Advantages:

- Takes into account dependence between variables

Limits:

- Adds a new hyperparameter k, hard to evaluate

K-nearest neighbor: example

Question

Impute missing values using:

- Using unique values: mean, median ...
- For $k = 3$ and $k = 1$, use the K-nearest neighbor algorithm to impute the missing value.

	Height	Weight
ID1	10	1
ID2	12	2.5
ID3	14	3
ID4	9	2
ID5	N/A	1

Feature scaling

Feature scaling

Feature scaling

Feature scaling* is a method used to normalize the range of features.

Feature scaling

Feature scaling

Feature scaling* is a method used to normalize the range of features.

Feature scaling can be useful in the case of:

- Algorithms that make assumptions regarding feature distribution

Feature scaling

Feature scaling

Feature scaling* is a method used to normalize the range of features.

Feature scaling can be useful in the case of:

- Algorithms that make assumptions regarding feature distribution
- Algorithms that take into account the values of the features (distance based)

Feature scaling

Feature scaling

Feature scaling* is a method used to normalize the range of features.

Feature scaling can be useful in the case of:

- Algorithms that make assumptions regarding feature distribution
- Algorithms that take into account the values of the features (distance based)
- Algorithms that use gradient descent

Min-max scaling

Min-max scaling

Min-max scaling* (rescaling) consists in rescaling the range of features to scale in the range [0, 1]:

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

Min-max scaling

Min-max scaling

Min-max scaling* (rescaling) consists in rescaling the range of features to scale in the range [0, 1]:

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

Question

Apply rescaling to the vector $[1, 3, 4, 2]$.

Standardization

Standardization

Standardization consists in transforming the feature to have zero-mean and unit-variance:

$$x' = \frac{x - \bar{x}}{\sigma}$$

with $\bar{x}$ the average and σ the standard error.

Standardization

Standardization

Standardization consists in transforming the feature to have zero-mean and unit-variance:

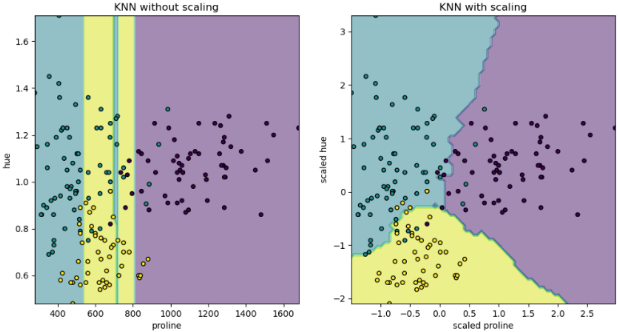
$$x' = \frac{x - \bar{x}}{\sigma}$$

with $\bar{x}$ the average and σ the standard error.

Question

Standardize the vector $[1, 3, 4, 2]$.

Example on Wine dataset



Great examples at: `https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importan`

Possible data leak

Data leaking between train and test set

Data leaking between train and test set consists in propagating information from the train set to the test set, rendering the results void.

Possible data leak

Data leaking between train and test set

Data leaking between train and test set consists in propagating information from the train set to the test set, rendering the results void.

Question

Why do you think feature scaling can cause data leak and how can you prevent it ?

Possible data leak

Data leaking between train and test set

Data leaking between train and test set consists in propagating information from the train set to the test set, rendering the results void.

Question

Why do you think feature scaling can cause data leak and how can you prevent it ?

Data leak between train and test dataset when scaling is a very frequent mistake. Be careful !

When should we scale features ?

When should we scale features:

When should we scale features ?

When should we scale features:

- Model sensitive to amplitude (distance based algorithms for example)

When should we scale features ?

When should we scale features:

- Model sensitive to amplitude (distance based algorithms for example)
- Gradient based algorithm (saves training time)

When should we scale features ?

When should we scale features:

- Model sensitive to amplitude (distance based algorithms for example)
- Gradient based algorithm (saves training time)
- When transforming variables

When should we scale features ?

When should we scale features:

- Model sensitive to amplitude (distance based algorithms for example)
- Gradient based algorithm (saves training time)
- When transforming variables
- When doing PCA

When shouldn't we scale feature ?

Feature scaling may not be a good idea in the case of:

When shouldn't we scale feature ?

Feature scaling may not be a good idea in the case of:

- Models we want to interprete

When shouldn't we scale feature ?

Feature scaling may not be a good idea in the case of:

- Models we want to interprete
- Some models do not care and simply take into account proportionality

Removing outliers

Outliers

Outliers

An outlier is a data point that **differs significantly from other observations**.

Outliers

Outliers

An outlier is a data point that **differs significantly from other observations**.

Outliers can be caused by:

- A measuring issue
- A variability in the measurement
- A **novel, unexpected behavior**

Why should we care outliers ?

Outliers can be:

- Due to a measuring error
- Due to the features and bear information . . . non-gaussian distribution for example !

Why should we care outliers ?

Outliers can be:

- Due to a measuring error
- Due to the features and bear information . . . non-gaussian distribution for example !

Dealing with outliers

Possibility to deal with outliers:

- Remove them from the dataset
- Replace the outlier value using imputation
- Use robust measuring metrics (median instead of mean)
- Adapt models accordingly

Why should we care outliers ?

Outliers can be:

- Due to a measuring error
- Due to the features and bear information . . . non-gaussian distribution for example !

Dealing with outliers

Possibility to deal with outliers:

- Remove them from the dataset
- Replace the outlier value using imputation
- Use robust measuring metrics (median instead of mean)
- Adapt models accordingly

Be careful before removal, as they can bear useful information !

Outlier detection using Tukey's fence

Tukey's fence

Tukey's fence is an usual method for outlier detection, that considers as outliers observations outside the range:

$$\left[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)\right]$$

Outlier detection using Tukey's fence

Tukey's fence

Tukey's fence is an usual method for outlier detection, that considers as outliers observations outside the range:

$$\left[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)\right]$$

Tukey suggests using $k = 1.5$ to flag individuals as *outliers* and $k = 3$ as *far-out*.

Questions

Questions ?