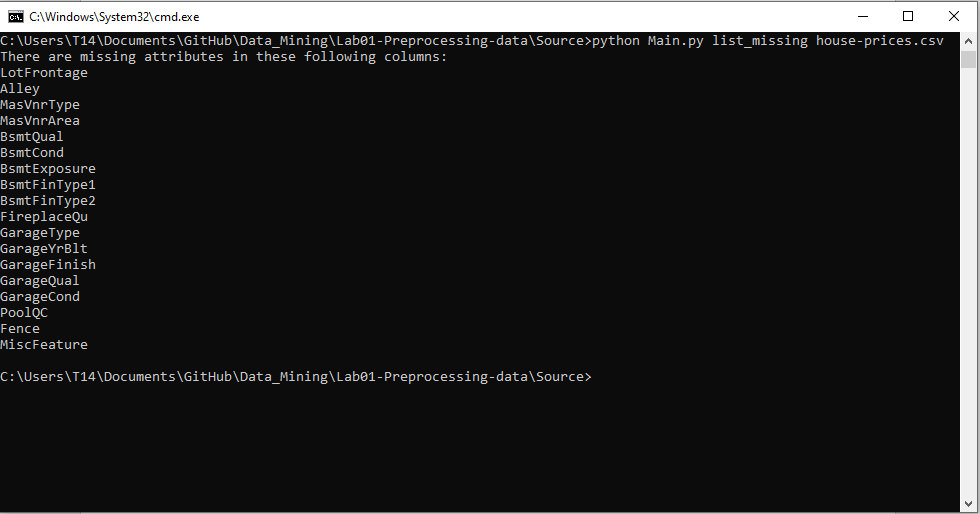
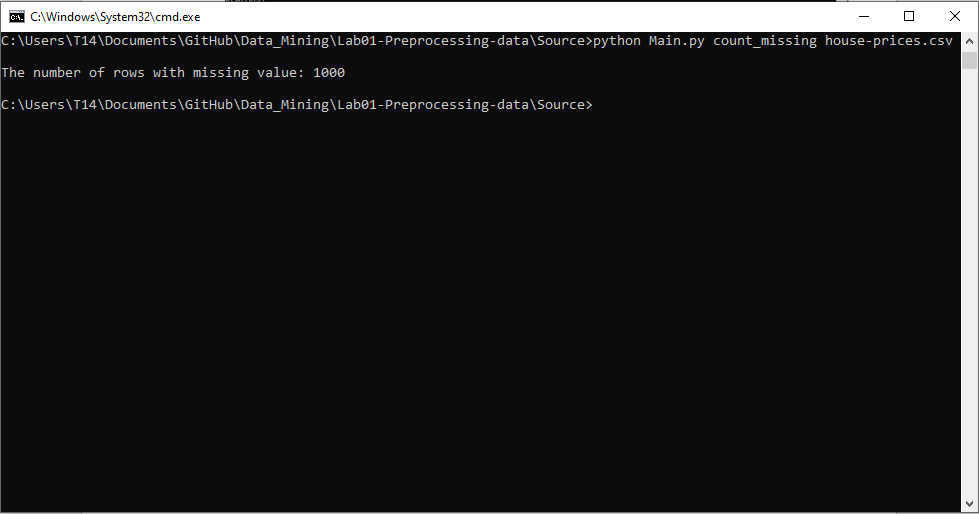
1. Extract columns with missing values

Run python command Main.py list\_missing house-prices.csv

The program lists the missing columns:

1. Count the number of lines with missing data.

Run python command Main.py count\_missing house-prices.csv,

The program prints out the number of missing values

1. Fill in the missing value using mean, median (for numeric properties) and mode (for the categorical attribute).

In function 3 use 3 options: column, method, outfile.

In there:

• column can take multiple parameters to indicate the columns to be filled in. If not specified, all columns will be filled by default. Columns with invalid names will be ignored.

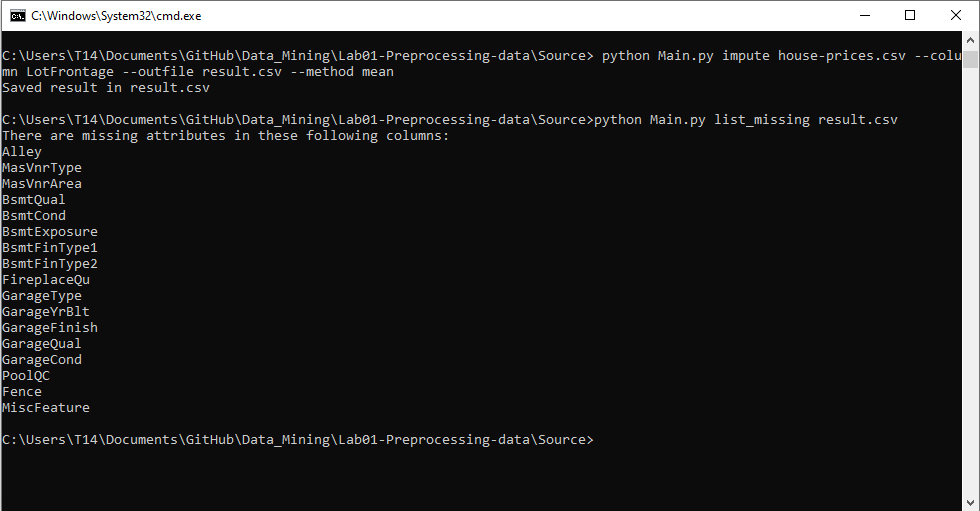
• method indicates the filling method for the numeric attribute: mean median and categorical: mode

• outfile indicates the name of the file to be saved

python Main.py impute house-prices.csv --column LotFrontage --outfile result.csv --method mean

– All missing LotFrontage values are replaced by its average

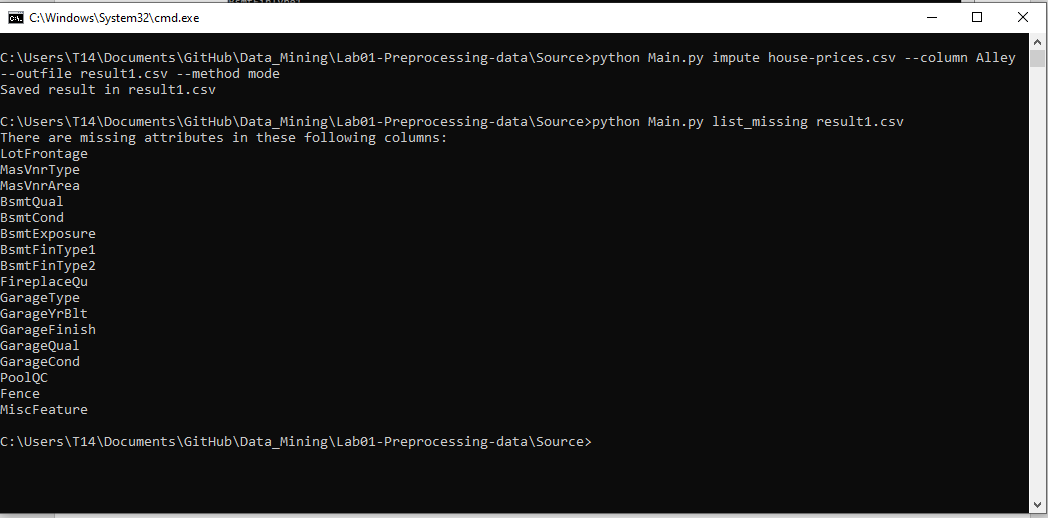
– Using function 1 on the result.csv file, the LotFrontage column is not found



python Main.py impute house-prices.csv --column Alley --outfile result1.csv --method mode

– All missing LotFrontage values are replaced by its mode

– Using function 1 on file result1.csv, the column Alley is not visible

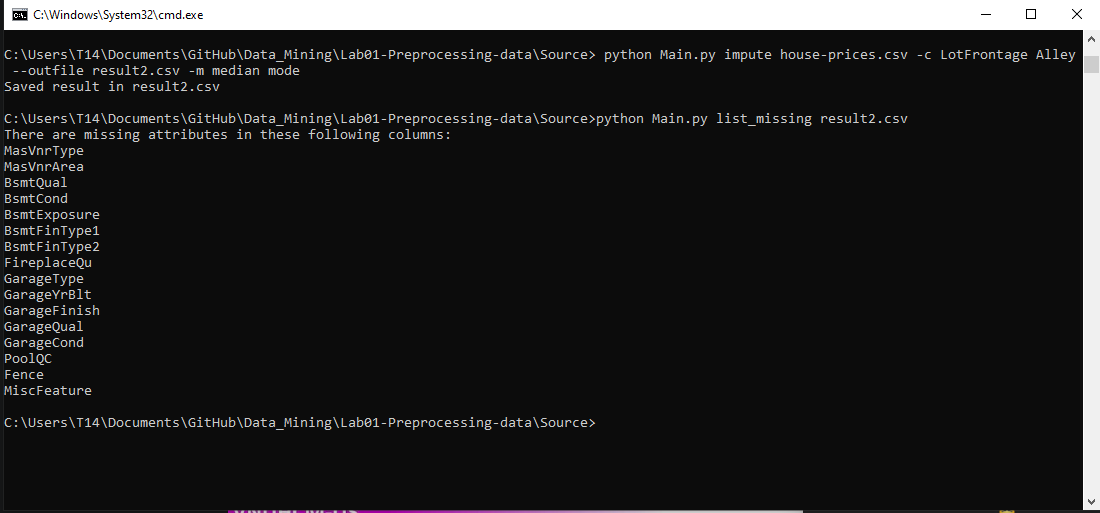


• python Main.py impute house-prices.csv -c LotFrontage Alley --outfile result2.csv -m median mode

– All missing LotFrontage values are replaced by its median

– All missing values of Alley are replaced by its mode

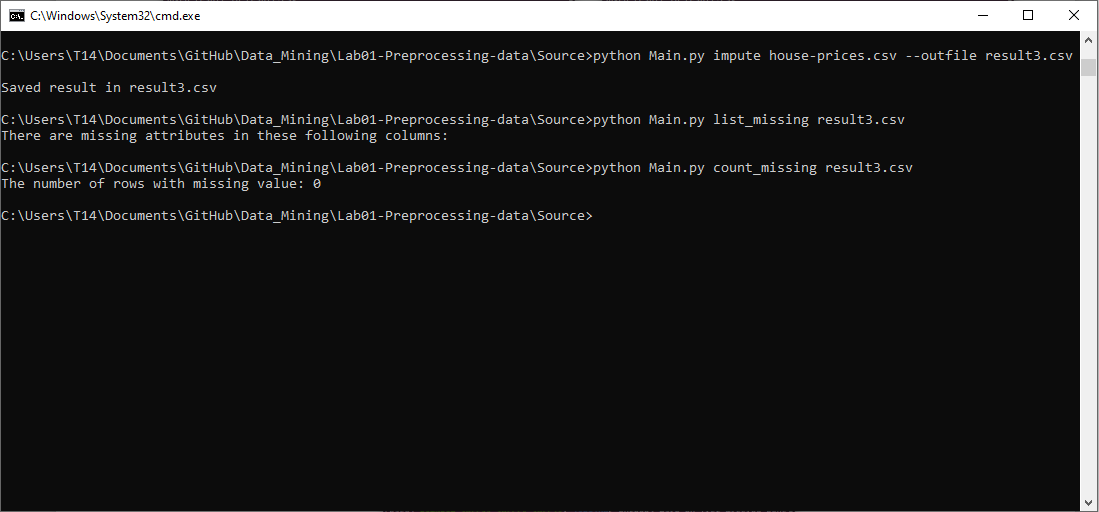
– Using function 1 on file result2.csv, the column LotFrontage, Alley is not visible



python Main.py impute house-prices.csv --outfile result3.csv

– All missing values in the table have been filled in

– Save to the file result3.csv

– Using function 1,2 does not detect missing values

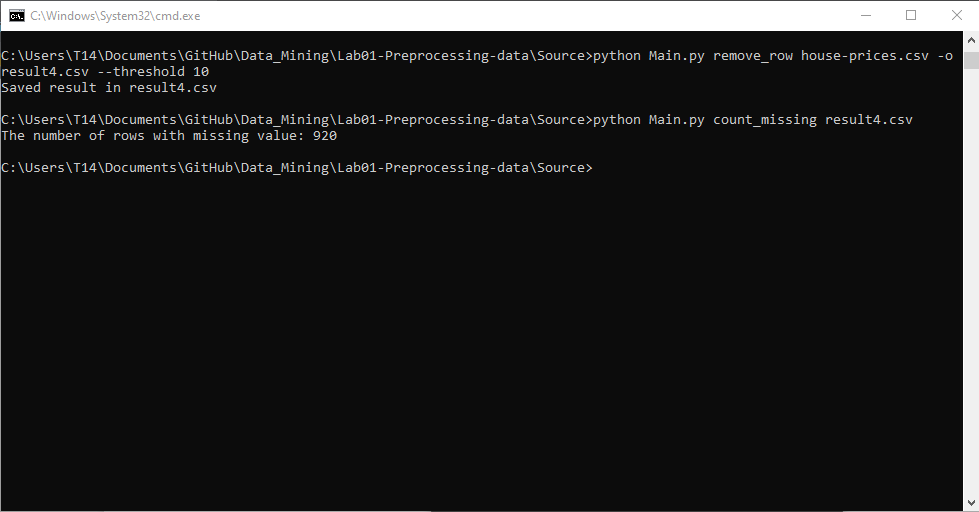
1. Deleting rows containing more than a particular number of missing values (Example: delete rows with the number of missing values is more than 50% of the number of attributes).

Function 4 needs 2 parameters, threshold, outfile

• threshold is the erase threshold, in percent, default is 50

• outfile indicates the name of the file to be saved

python Main.py remove\_row house-prices.csv -o result4.csv --threshold 10



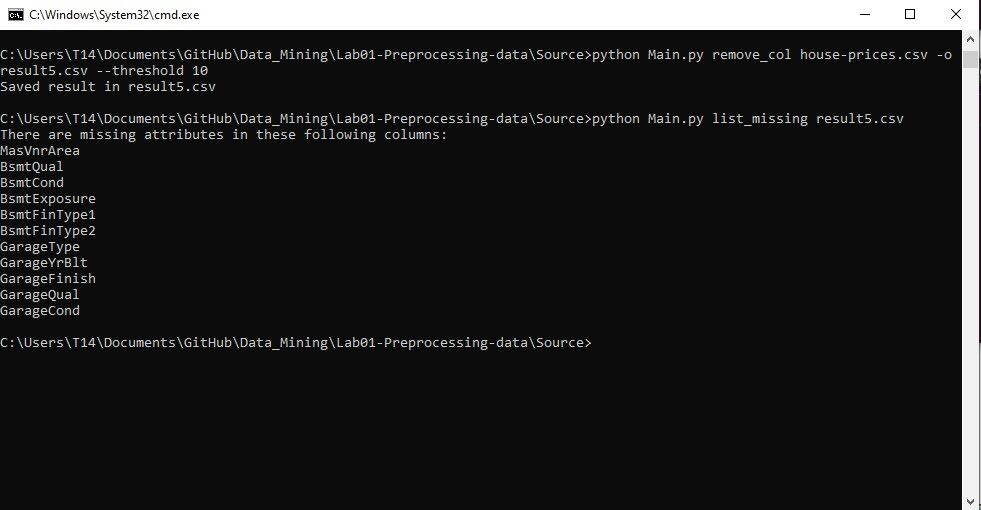
1. Deleting columns containing more than a particular number of missing values (Example: delete columns with the number of missing values is more than 50% of the number of samples).

Function 4 needs 2 parameters, threshold, outfile

• threshold is the erase threshold, in percent, default is 50

• outfile indicates the name of the file to be saved

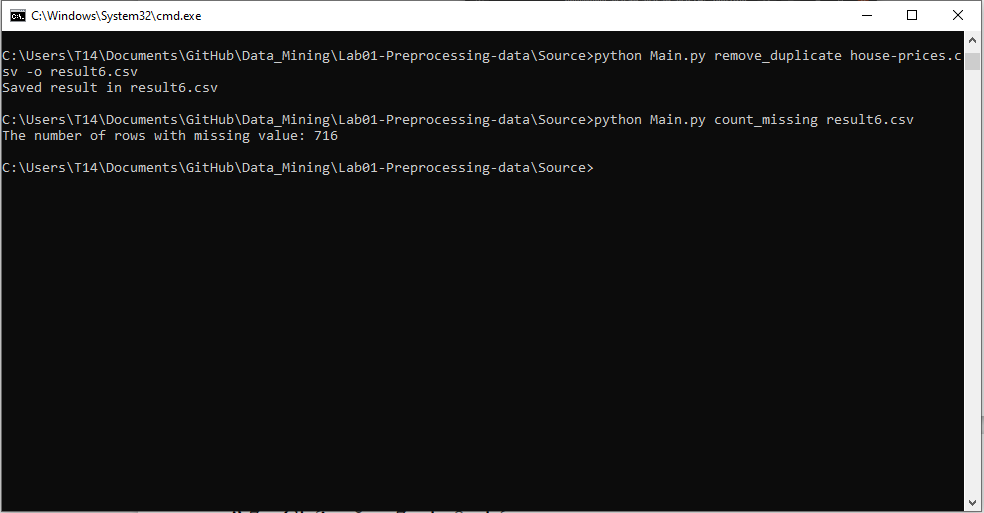
python Main.py remove\_col house-prices.csv -o result5.csv --threshold 10



1. Delete duplicate samples.

Function 6 needs 1 outfile parameter.

Run python Main.py remove\_duplicate house-prices.csv -o result6.csv



1. Normalize a numeric attribute using min-max and Z-score methods.

In function 7 use 3 options: column, method, outfile.

In there:

• column can take multiple parameters to indicate columns to normalize. If not specified, all columns are normalized by default. Categorical columns are ignored, invalid name columns are ignored

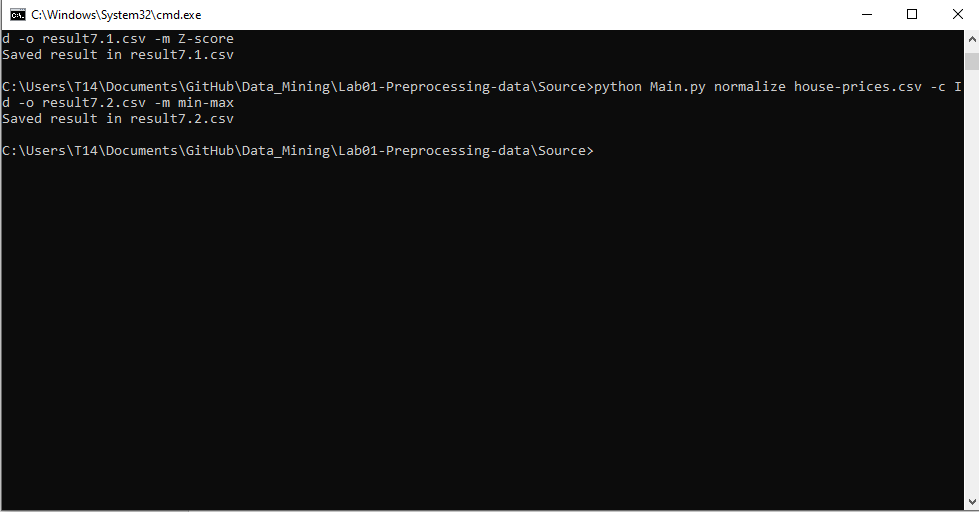
• method indicates normalization method for numeric attribute, supports min-max and Z-score, default is min-max.

• outfile indicates the name of the file to be saved, default is the current file.

With the following two commands, we have normalized the Id property in two ways:

python Main.py normalize house-prices.csv -c Id -o result7.1.csv -m Z-score

python Main.py normalize house-prices.csv -c Id -o result7.2.csv -m min-max



8. Performing addition, subtraction, multiplication, and division between two numerical attributes.

Function 4 needs 2 parameters, formula, outfile

• formula is formatted as =. For complex operations involving round brackets, the formula (or just the part after the equal sign) must be enclosed in double quotes.

• outfile same function as above.

python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id+MSSubClass

python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id-MSSubClass python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id\*MSSubClass python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id/MSSubClass

