How can I perform fuzzy matching in SAS, and can you provide an example?

Authored by

stats writer

June 23, 2024

Fuzzy matching is a technique used in SAS to compare strings of text and identify patterns or similarities between them. It is commonly used in data cleaning and merging processes to identify and correct spelling errors, typos, and other variations in data.

To perform fuzzy matching in SAS, you can use the Fuzzy Matching component in the Data Quality Transformation (DQT) process. This component uses algorithms and user-defined criteria to determine the level of similarity between strings of text.

For example, if you have a dataset with customer names that may have slight variations (e.g. "John Smith" and "Jon Smith"), fuzzy matching can be used to identify and group these names together. This allows for better data analysis and decision-making.

Additionally, SAS provides various functions and procedures such as the COMPGED and COMPLEV functions, and the FUZZY procedure, which can be used for more advanced fuzzy matching techniques.

In summary, fuzzy matching in SAS is a powerful tool for identifying and correcting data discrepancies, and it can greatly improve the accuracy and quality of data analysis.

Perform Fuzzy Matching in SAS (With Example)

Often you may want to join together two datasets in SAS based on imperfectly matching strings.

This is often called fuzzy matching.

The easiest way to perform fuzzy matching in SAS is to use the function along with the function.

Both of these functions are used to quantify the similarity between strings and can be used to "match" similar strings together.

The following example shows how to use these functions to perform fuzzy matching in SAS.

Example: How to Perform Fuzzy Matching in SAS

Suppose we have the following dataset in SAS that contains information about team names and points for various basketball players:

```
/*create first dataset*/
data data1;
input team $ points;
datalines;
Mavs 19
Nets 22
Kings 34
Warriors 19
Magic 32
;
run;
/*view dataset*/
proc printdata=data1;
```

Obs	team	points
1	Mavs	19
2	Nets	22
3	Kings	34
4	Warriors	19
5	Magic	32

And suppose we have another dataset with team names and assists for various basketball players:

```
/*create second dataset*/
data data2;
input team $ assists;
datalines;
Netts 8
Majick 7
Keengs 8
Warriors 12
Mavs 4
;
run;
/*view dataset*/
proc printdata=data2;
```

Obs	team	assists
1	Netts	8
2	Majick	7
3	Keengs	8
4	Warriors	12
5	Mavs	4

Notice that many of the team names in this dataset are similar but not exactly the same as the team names in the previous dataset.

We can use the following syntax in SAS to perform fuzzy matching and join together these two datasets based on similar team names:

```
/*use fuzzy matching to merge datasets based on similar team names*/
data data3;
set data1;
tmp1=soundex(team); /*encode team names from data1*/do i=1 to nobs;
set data2(rename=(team=team2)) point=i nobs=nobs;
tmp2=soundex(team2); /*encode team names from data2*/
dif=compged(tmp1,tmp2); /*determine similarity
```

```
between team names*/if dif<=50 thendo;
drop i tmp1 tmp2 dif; /*drop unnecessary variables*/output;
end;
end;
run;

/*view resulting dataset*/
proc print data=data3;
```

Obs	team	points	team2	assists
1	Mavs	19	Mavs	4
2	Nets	22	Netts	8
3	Kings	34	Keengs	8
4	Warriors	19	Warriors	12
5	Magic	32	Majick	7

The SOUNDEX and COMPGED functions are able to match team names based on similarity and produce one final dataset that merges the two datasets together.

The following tutorials explain how to perform other common tasks in SAS: