T.R.
MARMARA UNIVERSITY
INSTITUTE OF SOCIAL SCIENCES
DEPARTMENT OF SOCIOLOGY

Data Imputation with Deep-Learning Models

Final Project Report

Advanced Statistical Analysis – SOI8023

Lecturer
Asst. Prof. Mehmet Fuat KINA

Belkıs YÜCE
790824001

24.01.2025

1. Introduction

Missing data is a common problem encountered in almost all data sets and sometimes causes data analysis to fail. The problem I experienced due to missing data in an analysis I conducted in my master's thesis also led me to search for new methods on how to handle missing data. While trying to solve my missing data problem on the one hand, producing a final project for this course on the other hand led to this study: data imputation with deep learning. In this report, first I will explain the problem, then introduce the data and the variable used for the project. Then, I present the deep learning models that I used to solve the problem and interpret the results of the models. Finally, I will mention which new models can be tried in the future.

2. Research Topic and Data

This study used data from the Turkish Faith and Religiosity Survey (TFRS). The survey was conducted in 35 provinces across Turkey between December 2021 and May 2022 with 1942 people aged 18 and over. The survey includes approximately 200 questions on various topics such as religion and religiosity, freedom of belief and lifestyle, identities, perception of gender, religious practices, dimensions of faith, and happiness. A group of questions in the survey asks participants how close or distant they feel to certain identity categories.

Participants were asked to give a score for each identity category on a scale from "1 (the most distant)" to "5 (the closest)" with the question "*Please indicate how close you feel to the following categories*". The identity categories are "Agnostic," "Alevi," "Atatürkist," "Atheist," "Conservative," "Deist," "Democrat," "Environmentalist," "Feminist," "Humanist," Islamist," "Laic," "Leftist," "LGBT+ rights advocate," "Muslim," "Nationalist," "Religious," "Rightist," "Shiite," "Socialist," and "Sunni" (Faith and Religiosity in Turkey, 2023).

In my thesis, I developed an affective polarization analysis based on 21 identity variables of TFRS data, but the analysis results deviated from the literature at some points. When I examined the findings in-depth, I saw that the number of respondents was low for some variables, meaning that the missing data was high, which caused distortions in the analysis results. In order to solve the problem in the analysis, I had to handle the missing data. Therefore, in this study, I tried to impute the relevant variables with deep learning, a method used to predict and fill in missing data cells by learning the patterns of large and complex data sets (Yoon et al., 2018; Gondara and Wang, 2018).

3. Application

Before starting the coding part of the study, I researched deep-learning models suitable for my variable type. After learning a few models, I moved on to the coding part but also searched for new models according to the problems that arose during the process.

Target Variable Selection

There are 21 variables included in my analysis that have missing data problems. However, instead of working with all of these variables, I found it appropriate to apply the model by choosing a variable suitable for imputation and applying the same process to the other variables if it is successful. In order to decide which variable to choose, I first examined the missing data rates of my 21 identity variables. Since I knew that there was no missing data in the gender variable (q0001), I calculated the percentage of missing data of the 21 identity variables.

In order for the learning to be successful, there should not be too much missing data, so I decided to choose the variable "Humanist (q0050_0011)" as the variable with the highest but also with less than 50% (27.87%) missing data. However, seeing that the learning was not

successful, I thought that that there was not enough related variable in the data for learning. Therefore, I preferred to work on the variable of "Muslim (5.28% missing data)".

Feature Selection

I chose to do Lasso analysis for feature selection. In order to perform Lasso analysis, the number of observations in my target (the variable the model is trying to predict) and predictor (the variables the model will use to predict the target) variables had to be equal. While equalizing the numbers in target and predicyors, I had to keep enough observations to ensure that the model learned well and also include as many and diverse variables as possible in the analysis to provide better predictions. My data could currently provide representation at the NUTS-1 level and in order not to lose this, I first calculated the minimum sample size I needed to have in order to provide national representation at a 95% confidence interval. Accordingly, when I removed the missing data in all my variables, my sample size had to be around 800. I manually eliminated some of my predictor variables with the most missing data, so reached an equal number of full observations for all my variables and determined the optimum sample size as 884.

In the next step, I performed a Lasso analysis with the remaining predictor variables and determined the variables I needed to include in the model. I started my analysis with a penalty coefficient of 0.01 but later updated it to 0.1 due to the underfitting problem. I split the data into 70% for training and 30% for test. I obtained a Lasso score (R^2) of 0.58 and a mean squared error of 0.59. As a result of the analysis, I filtered the variables whose Lasso coefficients were not equal to zero to include them in the model. Accordingly, 17 variables that measure attitudes and perceptions on various topics were selected to be included in the deep learning models. I did cross-validation by dividing the data into 5 parts to see if the Lasso score I obtained gave similar results in different subsets. I concluded that it gave consistent results since there was not much difference between the highest R^2 (0.65) and the lowest R^2 (0.49) from the results.

Models for Likert Variable

In the next step, I tried the models that possibly give the best results for deep learning. Since the responses of my variable were in Likert type, I had to use a model that would make predictions while preserving the ordinal structure. Since the first models I tried did not give the performance I wanted, I had to try multiple different models and at one point convert my variable to categorical. In this part of the report, I will explain the models I tried and interpret the results.

Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a type of fully connected neural network and a foundational architecture in deep learning. It consists of an input layer, one or more hidden layers, and an output layer. Each neuron applies a weighted sum of its inputs, adds a bias, and passes the result through an activation function (e.g., ReLU, sigmoid, or tanh). They are widely used for classification, regression, and time-series tasks (Goodfellow et al., 2016).

Since I was working with an ordinal variable, I thought that the MLP model would be suitable for my variable structure. In order to ensure that the model learns better, I started the model with a layer of 256 neurons and put many hidden layers. While using "relu" activation in the hidden layers, since I wanted my output to be in the form of a 5-point Likert, I added a 5-neuron output layer with "softmax" activation suitable for the ordinal structure. In this model, I used

"adam" as the optimizer[1] and "mean squared error" as the loss function[2]. I started with 50 epochs (each pass of the model through the entire training data set) and then I increased the number of epochs to 100 because there was an underfitting problem in my model. I used the early stopping technique to stop when model learning do not increase any further. To evaluate this model's performance, I created a classification report and evaluated the precision[3], recall[4], and F1 scores[5].

Table 1: Classification Report of MLP (adam/mse)

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	97
2.0	0.00	0.00	0.00	60
3.0	0.00	0.00	0.00	204
4.0	0.00	0.00	0.00	186
5.0	0.70	1.00	0.82	1281
accuracy			0.70	1828
macro avg	0.14	0.20	0.16	1828
weighted avg	0.49	0.70	0.58	1828

According to the report, my accuracy F1 score shows that my model can predict with 70% accuracy. While the macro average value remains low at 0.16, the weighted average calculated by taking into account class imbalances is 0.58. Although this is not a bad result, my precision and recall values are 0 in response categories of 1,2,3,4, while the precision of category 5 is 0.7 and the recall value is 1. This shows that all my missing cells are predicted as 5.

Since this model did not achieve the success I wanted, I tried again by updating the MLP model with a different optimizer and loss function. I used the same number of neurons, number of layers, and activations in the new model. However, this time I used the "RMSprop" optimizer, which can be more sensitive to the ordinal variable structure. Since there was an imbalance between the response categories of my variable, I preferred to use the "categorical_crossentropy" loss function, which is more functional in imbalanced data sets. I kept the train-test split ratio and epoch number the same as the previous model.

[1] Algorithm used to update model weights and biases (Kingma & Ba, 2014).
[2] It is a function that tries to minimize the difference between the model's predictions and the actual values by measuring this difference (Goodfellow et al., 2016).
[3] A metric that measures to what extent positive predictions are truly positive (Goodfellow et al., 2016).
[4] A metric that measures how accurately the model predicts true positives (Goodfellow et al., 2016).
[5] It is the harmonic mean of precision and recall and is used to balance the classification performance. The value of F1-Score varies between 0 and 1, with 1 representing the best and 0 representing the worst performance (Goodfellow et al., 2016).

Table 2: Classification Report of MLP (RMSprop/categorical_crossentropy)

	precision	recall	f1-score	support
1	0.00	0.00	0.00	29
2	0.00	0.00	0.00	17
3	0.00	0.00	0.00	64
4	0.00	0.00	0.00	67
5	0.68	1.00	0.81	372
accuracy			0.68	549
macro avg	0.14	0.20	0.16	549
weighted avg	0.46	0.68	0.55	549

According to Table 2, accuracy F1 (0.68), macro average (0.16), and weighted average (0.55) scores have decreased compared to the scores of the previous model. As in the previous model, only category 5 was predicted in this model and the other categories were not predicted at all. However, similar to the decrease in other scores, the precision score for category 5 in this model has decreased to 0.68.

Autoencoder

Autoencoder is an artificial neural network architecture that is one of the unsupervised learning methods. It encodes the input into a compressed representation (latent representation) and then learns to reconstruct the input from this. It is usually used to reduce the data size or to understand the underlying structure in the data (Goodfellow et al., 2016; Hinton and Salakhutdinov, 2006).

Since the MLP model did not give the performance I wanted, I tried Autoencoder, a model used to better understand complex structures. This model provides learning by generating input and output in a part of the data and makes predictions according to the in-learning. The model starts with 128 neurons and aims to create an output with 5 neurons. It used “relu” in the classifier layers and “softmax” in the output layer. In this model, I preferred to use “adam” as the optimizer and “categorical_crossentropy” as the loss function. Similarly, I designed the model to be 100 epochs with early stopping. I determined the learning rate as 0.001 to provide slower but accurate learning, thus trying to solve the underfitting problem.

Table 3: Classification Report of Autoencoder (adam/categorical_crossentropy)

	precision	recall	f1-score	support
1	0.00	0.00	0.00	29
2	0.00	0.00	0.00	17
3	0.00	0.00	0.00	64
4	0.00	0.00	0.00	67
5	0.68	1.00	0.81	372
accuracy			0.68	549
macro avg	0.14	0.20	0.16	549
weighted avg	0.46	0.68	0.55	549

Table 3 shows that there was no improvement in the performance of my model. While the Accuracy F1 score was 0.68, the macro average was 0.16 and the weighted average was 0.55. Similarly, the recall score of the 5th category was 1 and the precision score was 0.68. The model also failed to predict categories other than the fifth category.

Models for Dichotomous Variables

When the models could not give enough performance for the Likert-type variable, I decided to simplify the model by making the variable dichotomous. I coded the Muslim identity as 1 and 2 “not Muslim” and 4 and 5 “Muslim” on SPSS. Although category 3 represents the neutral, it had to be included in a category for the variable to become dichotomous. Since the number of people who responded to category 5 was too high and created an imbalance, I coded category 3 as “not Muslim” in order to balance the data.

From now on, I repeated the same procedures I did for Likert models. I determined the optimum number of variables and sample size. In this case, the sample size was 1129. The target variable was “Being or not being Muslim (q0050_0001_d)”. In the Lasso analysis, I used a penalty coefficient of 0.01 again and obtained a Lasso score of 0.49. The mean squared error was 0.024. As a result of the Lasso analysis, 22 variables with a Lasso coefficient different from zero were selected as predictors.

Binary Classification Multilayer Perceptron

Since I now have a dichotomous variable, I first tried the Binary classification multilayer perceptron model. In the model I started with 256 neurons, I used “relu” for activation and “sigmoid” since my output would be binary. This time I set my learning rate as 0.01. I preferred “adam” as the optimizer and “binary_crossentropy” as the loss function. Similarly, I used epoch 100 in the data I divided as 70% for training and 30% for test.

Table 4: Classification Report of Binary Classification Multilayer Perceptron

(adam/binary_crossentropy)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	97
1	0.95	1.00	0.97	1731
accuracy			0.95	1828
macro avg	0.47	0.50	0.49	1828
weighted avg	0.90	0.95	0.92	1828

As seen in Table 4, I achieved a high F1 score of 0.95. The macro average was 0.49, and the weighted average was 0.92. The “Muslim (1)” category has a precision of 0.95 and a recall of 1. However, despite these scores, all predictions were made as “Muslim (1)” and the model did not work successfully. When I tried the same model with 64-neuron initialization, “softmax” activation in the output layer, and “mse” loss function but I obtained the same results.

Table 5: Classification Report of Binary Classification Multilayer Perceptron (adam/mse)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	97
1	0.95	1.00	0.97	1731
accuracy			0.95	1828
macro avg	0.47	0.50	0.49	1828
weighted avg	0.90	0.95	0.92	1828

As a result, all five models I tried did not give the results I wanted and predicted all the missing data in a single category. When I examined the models that could make more accurate predictions, I saw that there were studies (Collier et al., 2023; Carpita and Manisera, 2011) showing that DNN (Deep Neural Network) and ABBN (Approximate Bayesian Bootstrap) models are suitable for variables with ordinal structures. In my future studies, I will try the imputation again with new models.

4. References

Carpita, M., & Manisera, M. (2011). On the imputation of missing data in surveys with Likert-type scales. *Journal of Classification, 28*(2), 301-319. https://doi.org/10.1007/s00357-011-9074-z

Collier, Z. K., Kong, M., Soyoye, O., Chawla, K., Aviles, A. M., & Payne, Y. (2023). Deep learning imputation for asymmetric and incomplete Likert-type items. *Journal of Educational and Behavioral Statistics, 49*(2), 1-27. https://doi.org/10.3102/10769986231176014

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.

International Institute of Islamic Thought (IIIT). (2023). *Faith and religiosity in Türkiye*. Retrieved from https://iiit.org/wp-content/uploads/Turkish-Faith-and-Religiosity-in-T%C3%BCrkiye.pdf

Kingma, D.P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR, abs/1412.6980*.

Yoon, J., Jordon, J., & van der Schaar, M. (2018). GAIN: Missing Data Imputation Using Generative Adversarial Nets. *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Retrieved from https://www.vanderschaar-lab.com/papers/ICML_GAIN.pdf