Day2_Notes

1. What makes a good feature?
2. Feature Engineering for Machine Learning - Data Science Primer
3. FES, Chapter 8, Handling Missing Data
4. Examine, Delete & Impute Data Using Python:
    a. MLPR, Ch. 4, Missing Data (code: link)

Discussion:

Question:
A data set containing 150,000 chronological observations consists of a total of 22 numeric and categorical attributes. While exploring the data set you discover that 6,741 observations are riddled with missing data values. The missing data values are scattered throughout 2 numeric and 3 categorical attributes and are not specific to any particular chronological period.
Of the affected numeric attributes, one is integer based while one is continuous. For the affected categorical attributes, 2 contain ordinal classifications while one contains a nominal classification.

Briefly describe in the forum how you would address the missing data values (remember: most machine learning algorithms cannot be applied to data containing missing data values).
Be sure to explain why your approach would be effective and why it would be preferable to other possible approaches.

Solutions:

Answer 1:
Almost 5% (6741/150000) of the data is riddled with missing values within 5 attributes. One option can be to get rid of that record or delete the entire feature, but we might lose important information in the other 17 features that do have valid values, impacting the dataset. It would require a deeper understanding of the data and the results of deleting these records or attributes. I would not remove predictors that I believe have a high predictive power.

If we decide to retain all data, and considering most machine learning algorithms can't work with missing values, I would apply the following techniques:

Before making any changes, I would create a copy of the current dataset and perform some EDA to see, for example, the probability density function (PDF) of the numeric features. It would also help for understanding the nature and severity of missing information throughout the data.

For the 2 numeric attributes, and to avoid changing the PDF, I would create a regression model to calculate the appropriate value based on the rest of the non-missing features in each of the 6741 records to fill in the missing data. The specific type of model would be determined based on the relationship with the other predictor variables. In most cases this would work better than other approaches (like using mean, median or mode) since these techniques would likely alter the PDF impacting the rest of the data.

For the categorical attribute with nominal classification, we can create a value called “No followed by the feature name”. For example, assuming it is related to a specific odor then the missing values will turn into “No odor”. After imputing the new value, I would create dummy variables for each of the different variable values. This approach would be effective since we might turn to have a highly predictive feature related to the specific response variable in a ML model.

Lastly, for the 2 categorical attributes that contain ordinal classifications we can use algorithms like K Nearest Neighbors or multiple imputation (MICE algorithm) to predict the missing value based on the rest of the attributes in the record. This could be successful since the models can use all the other 17 non-missing attributes values to estimate for these 2 missing categorical ones. After imputing the values I would create the corresponding dummy variables. One of the benefits of these models is that we can tune different parameters to get a better estimate of the missing data depending on the characteristics of the dataset.

Answer 2:

The data set contains 150,000 observations, out of which 6,741 have missing data values. That means 4.5% of the observations have missing data values, and simply deleting those observations seems like a loss of a lot of data.

Addressing missing data in the numeric attributes:

Integer based: A simplistic way to impute missing integer values would be by replacing them with the median value. A better way would be to find which variables without missing values have a strong linear relationship with the Integer based variable, and use them as predictors in a linear regression model (rounding the predicted result). Another way would be to use a K-NN model, and impute the missing value with the median or mode of the known attribute values in K neighbours.

Continuous: A simplistic way to impute missing integer values would be by replacing them with the mean value. A better way would be to use a linear regression model (as described above) to

predict the missing values. Another way would be to use a K-NN model, and impute the missing value with the mean of known attribute values in K neighbours.
Addressing missing data in the categorical attributes:

Ordinal classifications: A simplistic way to impute missing ordinal values would be by replacing them with the most frequent category of the variable. A better way would be to use a tree-based model that sequentially splits the data into categories to predict the missing category. Since the data is chronological, another widely used imputation method would be Last Observation Carried Forward (LOCF), or Next Observation Carried Backward (NOCB), but the use of these methods highly depends on the nature of the dataset.
Nominal classification: As in the Ordinal case, a simplistic way to impute missing values would be by replacing them with the most frequent category of the variable. A better way would be treating the missing data as if they were an additional category of the variable, labeling them "Missing".

Answer 3:
There are several important factors to consider here. While we have 5 attributes of data out of 22, we may first want to consider if the missing data is indeed random or if we see some specific inference in terms of content. That is, perhaps the missing values are in fact values that are important information. An example of that is when there are studies with patients, in which a patient that had side-effects from a drug, which rendered them incapable of completing a study ends up dropping out of a study, it is a very important point to record. As such, we need to check if the data is missing at random or it is due to a specific cause. This would generally require expert knowledge, something that may require either research or asking a person with domain specific knowledge.

A great tool to help in figuring out the nature of the missing data is to visualize it with co-occurrence plots and to check relationships with heat-maps. This allows us to check if the missing data is related across categories, and if the missing data is in fact part of a larger, more meaningful trend.

Our book also mentions using PCA to analyze missing data. If we transform our data into booleans w/r/t missing or not, we can determine the nature of this. Samples without missing data will be projected onto the same location close to the origin, while samples with missing data will be further away. Of course looking at the missing data through numeric summation also gives us insight into the categories and nature of what is missing.

Once an opinion is made about the data and the nature of the missingness, we can can attempt various approaches at building a model. This includes deleting data, which has several obvious drawbacks, of encoding pertinent missingness, such as the "no alley" example in the book, or imputing data, which may change the distribution and place another layer of fabricated complexity into our analysis - one that we cannot easily separate, as we do not know what the

true hidden distribution actually is. The book describes this as "a model embedded within another model", which is an excellent way to express the compounded difficulty created.

Due to the very nature of our data, the continuous data may need to be imputed with a median or the like, however, the integer based attribute would do better to use a k-means clustering to categorize its likely position and impute it that way.

Nominal categories can also be difficult to place, as without a means to classify them (which as the book suggests is possible but not accurate with a k-means clustering or potential linear model searching for similarities in other respects) one may instead look to see if a level of feature engineering is feasible. That is, perhaps domain knowledge allows us to drop a category because it is so highly correlated with another that it only detracts from the model, or perhaps one category bleeds so heavily into another that reasonable inferences can be made. So, for instance, if we create a regression on two categories and found a tight relationship, we may be able to "guess" a variables nominal or ordinal position. It wouldn't be surprising to find that using feature engineering such as dividing X by Y yields a regular pattern that leads us to fill in the data with some bound of certainty.

We could of course create multiple models or work with certain types of trees that can handle some missing data, but we first may want to check if an attribute is in fact reasonably related to our analysis or logically extraneous, and furthermore, we may want to check if a variable might just be categorized better and then our missing values can retain one of those new induced categories.

Of course, with integers we can always create a category called zero, despite the possible noise we are introducing into the data set, however, with nominal and ordinal categories without some insight into the domain or some insight into the structure of the data, it is quite difficult to have a solution which will be fully valid.

On the other hand, if the data is scattered randomly, 6000/150000 is .04 which perhaps could mean that even a reasonable choice of inputs would not entirely skew things, especially if it were mixed between the training and validation sets.

So, in short, the first step is to get deep domain specific knowledge. The second step is to look for possible patterns to be exploited with a basic EDA. And the third step is to compromise on a process that may introduce noise, but does not disrupt our ML outcomes significantly.