Data Preparation, Analysis and Manipulation:

1. **Introduction:**

The standard data cleaning activity to prepare dataset for Machine Learning based prediction algorithm consists of primarily 2 steps as depicted below:

Raw Data

Technically Correct Data

Consistent Data

Type Checking and normalization

Fix and Impute

Fig-1: The 2-step data cleaning process (Ref: Jonge & Loo)

1. **Data Cleaning Approaches for the Customer Churn Dataset:** 
   1. Validity Checking and Data Imputation:

Typically, for any ML based prediction framework, there can be any number of validity rules used to clean the data, and these rules will depend upon the intended purpose or objective of the model. A summary of our data set is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. of Columns | No of Rows | No. of floating-point variables | No of Integer variables | Other |
| 100 (including the predicted variable - Churn) | 100000 | 56 | 22 | 3 |

**Summary of Missing Values:**

|  |  |  |  |
| --- | --- | --- | --- |
| Total No of Columns having Missing values | Total count of missing values | % of Missing Values w.r.t the total dataset | No of columns that do not have any missing values |
| 26 | 157757 | 1.57% | 74 |

|  |  |  |
| --- | --- | --- |
| Fields | No. of missing values | % of total no of rows |
| numbcars | 49366 | 49.36% |
| lor | 30190 | 30.10% |
| income | 25436 | 25.43% |
| adults | 23019 | 23.01% |
| hnd\_webcap | 10188 | 10.18% |

**Top-5 columns with highest frequency of missing values**

* **Rule based validation:** Since we do not have much clarity on the significance of each of the fields, we could not create a valid set of rules for rule-based consistency check. However, we checked following:

**Type checking and consistency of the fields**: No field with inconsistent data type was found. Values across the columns were consistent in the sense that there was no type mismatch between the values in the same column. However, we found almost 1.57% of the cell values as missing or ‘NA’. We imputed the missing values using various algorithm as explained in the next section.

**Data imputation for missing values:** Imputation is the process of estimating or deriving values for fields where data is missing. There is a vast body of literature available on various imputation techniques. For our data-preparation, we took scenario-based approaches for each field that had missing values:

* **Data Imputation for Numerical Variables:**

We adopted following approaches for numerical variables:

1. Random sampling from the set of most frequently occurring values: For the numerical fields that had integer values and possibly representing some counts or frequencies we imputed the missing values through a random sampling from the set of values with highest frequencies.
2. Max Value: For the fields that had one value with overwhelming frequency over the other values, we simply replaced the missing value with that particular value.
3. Mean: For the fields that have values distributed across a wide range and have floating-point values, we took the mean of the fields and imputed the missing values with the mean of the rest of the values for those fields.

* **Data Imputation for Categorical Variables:**

Frequency Based Approach: For the categorical fields, we simply replaced the missing values with the one that is having the highest frequency. In our case, for most of the categorical variables, we had one or two values that were having overwhelming frequency of occurrences against others and hence were the rational choice to use for imputation of the missing values.

* 1. Feature Engineering:

There’s a vast body of literature for feature engineering for datasets in order to apply ML algorithms.

Feature Engineering for Numerical Features include:

* Scaling:

Various scaling mechanism are in use for Machine Learning input dataset. The most commonly used ones are: Min-max normalization, Mean Normalization, Z-score normalization and Scaling to unit length.

For our dataset, we used min-max normalization:

* Transformations

The typical Transformation algorithms that are used for non-zero data are: logarithmic, square root transformation, and the multiplicative inverse. Also, there are options to transform data to follow a normal distribution, uniform or any other arbitrary distribution or variance stabilizing transformations. However, since, we do not have much business insights in our dataset, we have not applied any transformation as such to our dataset.

* Outlier clipping:

While our dataset had some fields that clearly shows asymmetric distribution, we have not used outlier clipping to allow sufficient noise in our input data to test the strength and performance of the classifier.

* Feature Interactions:

While some of the numeric fields in our dataset(especially the following fields: rev\_Mean, mou\_Mean, totmrc\_Mean, da\_Mean, ovrmou\_Mean, ovrrev\_Mean, vceovr\_Mean, datovr\_Mean, roam\_Mean) might have some correlation between them, since we do not have enough business insights into the dataset, we refrained from removing the columns with high correlations.

* Approximation of Addition and Multiplication:

We have not applied any additive or multiplicative approximation algorithm to the dataset as we have done some extensive data imputation to prevent any data sanity loss due to any addition-multiplication approximation.

* Discretization:

Most of our numeric and continuous variable were well distributed over a range, allowing not much scope for discretization.

There are several feature Engineering techniques available on categorical variables:

* Convert to numeric
* Label encoding
* Frequency encoding
* One-Hot encoding
* Weight of evidence
* Interaction features: min, max, avg & group by

As for Categorical features, we adopted one-hot encoding, to convert any categorical values into multiple columns having binary ‘0’ or ‘1’ and dropping the original columns. After one hot-encoding of all categorical fields, the total no. of fields (excluding the original categorical fields) increased from 100 to 250.

Reference:

1. Edwin de Jonge, Mark van der Loo: An introduction to data cleaning with R, Statistics Netherlands
2. <https://www.kdnuggets.com/2018/05/packt-tackle-common-data-cleaning-issues-r.html>
3. <https://blog.dominodatalab.com/manual-feature-engineering/>
4. <https://en.wikipedia.org/wiki/Feature_engineering>
5. https://en.wikipedia.org/wiki/Data\_transformation