Data Cleaning and Preprocessing Using R

Customer Churn Dataset (Kaggle)

April 6, 2025

1 Objective

This assignment demonstrates data cleaning, preprocessing, analysis, and visualization techniques using the "Customer Churn" dataset. The goal is to prepare and explore the dataset using various R packages, uncover hidden patterns, and prepare the data for modeling.

2 R Package Installation and Descriptions

The following R packages were used for preprocessing and analysis:

- `tidyverse`: A core package collection including `dplyr`, `ggplot2`, `readr`, useful for data manipulation and visualization.
- `GGally`: Extension of `ggplot2` that provides `ggpairs()` for visualizing relationships between variables.
- `ggcorrplot`: Displays correlation matrices visually in heatmap style.
- `plotly`: Interactive plotting tool to create dynamic 3D plots, helpful in visualizing PCA components.
- `factoextra`: Tools to extract and visualize multivariate data analysis results, especially PCA.
- `arules`: Implements algorithms for mining association rules and frequent itemsets.
- `seriation`: Techniques for reordering matrices or datasets to reveal underlying patterns.

- `sampling`: Provides sampling techniques including stratified sampling, which ensures balanced samples.
- `caret`: A unified interface for machine learning in R. Supports preprocessing, resampling, and training.
- `proxy`: Computes distances or similarities between observations, useful for clustering and proximity analysis.

3 Dataset Description

The **Customer Churn** dataset includes variables describing customer demographic information, tenure, service subscriptions, and whether they churned (left the company). Key features include:

- `tenure` (customer duration in months)
- `MonthlyCharges`
- `TotalCharges`
- `Churn` (Yes/No)

4 Data Preview & Summary

Using `print(churn, n=3, width=Inf)` and `summary(churn)`:

- Identified **missing values** in `TotalCharges`.
- Some variables had **skewed distributions**, indicating potential need for scaling or transformation.
- Data types included numeric, character, and logical/factor.

5 Mean of Numeric Variables

Used `summarise_all(mean)` on numeric features to understand central tendencies. Observed:

- Average tenure was approximately 32 months.
- Monthly charges varied widely, indicating pricing tiers.
- `TotalCharges` also varied depending on tenure and services used.

6 Feature Relationships with Pairwise Plot

Using `GGally::ggpairs()`, plotted numerical variables colored by `Churn`. Observations:

- Strong positive correlation between `tenure` and `TotalCharges`.
- Churned customers had lower tenure and slightly higher monthly charges.
- Visualization helped in identifying potential separability between churn classes.

7 Data Cleaning

```
clean.data <- churn %>% drop_na() %>% unique()
```

- Removed missing values and duplicates.
- Summary of `clean.data` showed improvement in data quality with more consistent statistics.

8 Aggregation by Churn Status

Grouped data by `Churn`:

- **Mean and median statistics** helped uncover:
  - Churned users had lower average tenure.
  - Higher average monthly charges.
  - Slightly lower total charges due to early dropout.

9 Random and Stratified Sampling

Used:

```
sample(c("Yes", "No"), size = 10, replace = TRUE)
```

for random sampling. Then performed **stratified sampling** using `sampling::strata()`:

- Ensured **equal representation** of `Churn = Yes` and `No`.
- Followed up with `ggpairs()` to visualize sampled data—distribution remained balanced.

10 Dimensionality Reduction – PCA

Performed PCA:

```
pc <- churn %>% select(-Churn) %>% as.matrix() %>% prcomp()
```

- Used `plotly::plot_ly()` for 3D scatter plot of tenure, monthly charges, and total charges.
- `plot(pc)` displayed variance explained by each principal component.
- First 2-3 PCs captured most of the variance, indicating dimensionality reduction was effective.

11 Visualizing PCA

- `churn_projected` (principal components) was plotted using `ggplot2`, revealing clusters.
- Visual separation in PC1 vs PC2 showed potential for use in classification tasks.
- `fviz_pca()` and `fviz_pca_var()` helped understand variable contributions to components.

12 Data Discretization

```
ggplot(churn, aes(x = MonthlyCharges)) + geom_histogram(binwidth = 5)
```

- Binned monthly charges to observe frequency distribution.
- Showed peaks around certain pricing tiers.

13 Distance and Similarity Measures

Used `proxy::dist()` for similarity analysis between customer records:

- Compared distances using **Euclidean**, **Manhattan**, and **Maximum** metrics.
- Found customer profiles with similar tenure and charges were closer in distance space.

14 Correlation Analysis

```
cor(MonthlyCharges, TotalCharges)
cor.test(MonthlyCharges, TotalCharges)
```

- Found a **strong correlation** between monthly and total charges (positively linear).
- Regression line confirmed the relationship.

15 Data Inspection & Sampling

```
take <- sample(seq(nrow(churn)), size = 15)
```

- Extracted random rows for inspection—confirmed consistency after cleaning.

16 Normalization

The normalization of numeric variables was performed using the following code:

```
churn_norm <- churn %>%
  select_if(is.numeric) %>%
  mutate_all(~ (.-min(.)) / (max(.) - min(.)))
```

This approach scales all numeric variables to a range between 0 and 1. After performing the normalization, the summary of the normalized data is as follows:

- The numeric variables have now been rescaled to a range of [0, 1].
- This is essential for machine learning models that rely on distance metrics, such as k-NN or PCA, to ensure no variable dominates due to its scale.

17 Feature Engineering

We created a new attribute, `HighCharge`, based on the `MonthlyCharges` column, which helps to classify customers based on their monthly charges.

```
churn <- churn %>%
  mutate(HighCharge = if_else(MonthlyCharges > 70, "High", "Low"))
```

- The new feature, `HighCharge`, labels customers with monthly charges greater than 70 as `High` and others as `Low`.
- This feature can help in segmentation tasks and understanding churn based on pricing tiers.
- Example output:

```
head(churn)
```

```
A tibble:  6 × 21
```

```
tenure MonthlyCharges TotalCharges Churn HighCharge
```

```
<dbl> <dbl> <dbl> <fct> <chr>
```

18 Conclusion

This analysis showcased a full pipeline of preprocessing techniques in R. From loading, cleaning, and sampling, to PCA and correlation analysis, the workflow effectively prepares data for downstream machine learning tasks like churn prediction. The visualizations and transformations provided a solid understanding of the dataset structure and key differentiators for churn behavior.