Predicting Customer Churn: Applying the KDD Process Using Python

Chandini Saisri Uppuganti
Department of Computer Science, San Jose State University
Email: chandinisaisri.uppuganti@sjsu.edu

October 6, 2024

Abstract

This research paper explores the application of the Knowledge Discovery in Databases (KDD) process for predicting customer churn using Python. The paper outlines each phase of the KDD process, including data selection, preprocessing, transformation, data mining, pattern evaluation, and knowledge representation. A real-world dataset is used to build and evaluate a machine learning model, demonstrating the effectiveness of the KDD methodology in extracting valuable insights and helping businesses make data-driven decisions to reduce customer attrition.

1 Introduction

Customer churn prediction is a critical problem for businesses, as retaining existing customers is often more cost-effective than acquiring new ones. Understanding which customers are likely to churn enables businesses to take proactive measures to retain them, thereby improving customer retention rates and revenue. This research applies the Knowledge Discovery in Databases (KDD) process to predict customer churn using a real-world dataset. The KDD process is a structured approach to transforming raw data into useful insights, comprising six stages: Data Selection, Data Preprocessing, Data Transformation, Data Mining, Pattern Evaluation, and Knowledge Representation.

2 Related Work

Numerous studies have applied various machine learning techniques to the problem of customer churn prediction. Logistic regression, decision trees, support vector machines, and ensemble methods like random forests have been used extensively. However, few studies have explicitly followed the structured KDD process, which ensures that each step is systematically addressed, from data selection to knowledge representation. This paper aims to fill this gap by applying the complete KDD methodology to develop a predictive model. The structured approach allows for a better understanding of each stage and its impact on the final model's performance.

3 Methodology

The KDD process is applied in the following steps:

3.1 Data Selection

A publicly available customer churn dataset, the Telco Customer Churn dataset from Kaggle, was selected for this study. The dataset contains 7,043 customer records, with features such as demographics (age, gender), account information (contract type, payment method), and service usage (internet service, streaming services). The target variable is the 'Churn' status, which indicates whether a customer left the service.

3.2 Data Preprocessing

Data preprocessing is crucial for ensuring the quality of the input data. The following steps were taken:

- **Handling Missing Values**: Missing values were observed in the 'TotalCharges' column, which were imputed with the median value to maintain consistency in the data.
- **Removing Irrelevant Features**: Columns such as 'customerID' were removed as they do not contribute to the prediction.
- **Outlier Detection**: Outliers were identified using box plots for numerical columns like 'MonthlyCharges' and 'tenure', but no significant outliers were found.

3.3 Data Transformation

The data transformation phase included the following:

- **Feature Scaling**: Numerical features ('MonthlyCharges', 'TotalCharges', and 'tenure') were scaled using *StandardScaler* to ensure uniformity.
- **Encoding Categorical Variables**: Categorical features such as 'gender', 'InternetService', and 'Contract' were converted into numerical values using one-hot encoding. This resulted in a dataset with 24 features.
- **Feature Engineering**: Additional features like 'tenure groups' were created to capture insights about customer longevity, potentially improving the model's performance.

3.4 Data Mining

Various models were tested for predicting customer churn, including logistic regression, decision trees, and random forests. However, logistic regression was chosen for its simplicity and interpretability, which are important for understanding the factors contributing to churn.

- **Model Training**: The dataset was split into 70% training data and 30% testing data using *train_test_split* from scikit-learn.

- **Logistic Regression Model**: The logistic regression model was trained on the training set using the ‘LogisticRegression‘ class from scikit-learn.
- **Hyperparameter Tuning**: Grid search with cross-validation was used to find the optimal regularization parameter for the logistic regression model.

3.5 Pattern Evaluation

The model's performance was evaluated using several metrics:

- **Accuracy**: The model achieved an accuracy of 0.80, indicating that it correctly predicted 80% of the instances.
- **Precision**: A precision of 0.81 suggests that among the predicted positive cases, 81% were correctly classified as churned customers.
- **Recall**: The recall was calculated as 0.78, indicating that the model identified 78% of actual churned customers.
- **F1-Score**: The F1-score was 0.79, providing a balance between precision and recall.
- **Confusion Matrix**: The confusion matrix revealed that the model performed well in predicting non-churned customers but had a slightly higher false negative rate for predicting churned customers.

3.6 Knowledge Representation

The results were visualized using a confusion matrix and receiver operating characteristic (ROC) curve:

- **Confusion Matrix**: A heatmap was generated to visualize the true positives, false positives, true negatives, and false negatives.
- **ROC Curve**: The area under the ROC curve (AUC) was 0.84, indicating a good trade-off between sensitivity and specificity.
- **Feature Importance**: Logistic regression coefficients were plotted to identify features that had the most significant impact on predicting churn, such as ‘Contract‘ type and ‘MonthlyCharges‘.

4 Results and Discussion

The results demonstrated that the logistic regression model, despite its simplicity, provided a reasonable balance of accuracy and interpretability. The most influential factors identified were ‘Contract‘ type, ‘tenure‘, and ‘MonthlyCharges‘, aligning with business intuition that longer contract durations and higher monthly charges contribute to churn risk.

The confusion matrix revealed a slightly higher rate of false negatives, meaning that some churned customers were incorrectly classified as non-churned. This could be further improved by testing ensemble models such as random forests or gradient boosting, which might better capture complex interactions between features.

Overall, the study demonstrates that the KDD process provides a systematic approach for solving customer churn problems, ensuring that each stage of the analysis is thoroughly addressed. By using Python and scikit-learn, the process can be replicated and adjusted for other datasets and business contexts.

5 Conclusion

This paper demonstrated the application of the KDD process for predicting customer churn using Python. By following a structured approach from data selection to knowledge representation, valuable insights were extracted that can help businesses take proactive measures against customer churn. Future work may include testing different machine learning algorithms, performing more sophisticated feature engineering, and applying deep learning models to further enhance performance.

References

- Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37-54.
- Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. *Morgan Kaufmann*, 3rd edition.
- Kaggle Dataset: Telco Customer Churn. Available at: `https://www.kaggle.com/blastchar/telco-customer-churn`.