**Random Forest Regression to Predict Customer Tenure**

Kelseyann Wright

Student ID: 00565814

College of Information Technology, Western Governors University

Data Mining I - D209 Task 2

Dr. Festus Elleh

June 29, 2023

**Random Forest Regression to Predict Customer Tenure**

**A1. PROPOSAL OF QUESTION**

The question that will be addressed is, "Can a random forest regression be used to predict telecom customer tenure?". Longer tenure means that the customers stayed with the same company for a longer period of time. This research question is important because predicting customer tenure will allow the company to take steps to increase tenure and improve business outcomes. In addition, identifying the factors that influence tenure will allow the business to implement targeted changes to increase customer tenure and retain customers for longer periods of time.

**A2. DEFINED GOAL**

The main goal of this analysis is to develop a random forest regression model to predict customer tenure using appropriate features.

**B1. EXPLANATION OF PREDICTION METHOD**

Random forest is a supervised learning algorithm that combines the strength of multiple decision trees through aggregation or “bagging” (Lyashenko, 2021). As an ensemble method using multiple unique decision trees with random feature selection, random forest is less prone to overfitting than a decision tree (Brieman 2001, Lyashenko, 2021). In addition, random forest runs fast, tends to work well without hyperparameter tuning, and can show feature importance (Lyashenko, 2021). These features make random forest a highly useful and robust technique for prediction analysis.

The random forest algorithm can be used for classification or regression. Random forest regression is used when the dependent variable is continuous. The goal of this project is to predict customer tenure, which is a continuous variable, making random forest regression an appropriate technique for our prediction.

**B2. SUMMARY OF METHOD ASSUMPTION**

One assumption of random forest regression is that your data sample is representative. This means that the data we are using is representative of the population, and therefore the patterns we see in the sample are likely to exist throughout the population. This is a common assumption in statistical analyses. Random forests are nonparametric, meaning the can be used with various data including both ordinal and non-ordinal categorical data as well as numeric data with non-normal distributions, even skewed and multi-modal data (Brieman, 2001).

**B3. PACKAGES OR LIBRARIES LIST**

The libraries used in the analysis are described in Table 1.

***Table 1: Justification of python libraries used in the analysis.***

|  |  |  |
| --- | --- | --- |
| Library | Justification / Use | Source |
| Pandas | Data frame manipulation and summarization | McKinney, 2010 |
| numpy | Data manipulation and formatting for statistical analyses | Harris et al., 2020 |
| Sci-kit learn | Includes a function for splitting data into train and test sets and a function for executing the Random Forest algorithm. This library also includes multiple functions for testing model accuracy. | Pedregosa et al., 2011 |

**C1. DATA PREPROCESSING**

One data preprocessing goal is to identify categorical features with high cardinality (greater than 5 groups) and remove them from the dataset. Random forest algorithms can be useful for identifying feature importance. However, feature importance is biased toward features with high cardinality. Although feature importance will not be addressed for the current research question, it may be useful later to determine which features are the best predictors of customer tenure. This is important information when making business decisions. Therefore, removing features now that may cause issues in later post-hoc analyses will streamline our ability to make data driven decisions later.

**C2. DATA SET VARIABLES**

Table 2: List of variables used in random forest analysis, and their data types.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Type** | **Variable** | **Type** |
| Latitude | Numeric, continuous | Internet Service | Categorical |
| Longitude | Numeric, continuous | Phone | Categorical |
| Area | Categorical | Multiple | Categorical |
| Income | Numeric, continuous | OnlineSecurity | Categorical |
| Marital | Categorical | OnlineBackup | Categorical |
| Gender | Categorical | DeviceProtection | Categorical |
| Churn | Categorical | TechSupport | Categorical |
| Outage\_sec\_perweek | Numeric, continuous | StreamingTV | Categorical |
| Techie | Categorical | StreamingMovies | Categorical |
| Contract | Categorical | PaperlessBilling | Categorical |
| Port\_modem | Categorical | PaymentMethod | Categorical |
| Tablet | Categorical | MonthlyCharge | Numeric, continuous |
| Tenure\* | Numeric, continuous | Bandwidth\_GB\_year | Numeric, continuous |

*\*Dependent variable*

**C3. STEPS FOR ANALYSIS**

Categorical variables with high cardinality, specifically more than five groups, were dropped from the dataset (see section C1). The remaining feature variables were transformed into dummy variables for the analysis. Dummy variables are binary values of 0 or 1, where 0 represents "no" or not being a member of that group and 1 represents "yes" or being a member of the group. Using dummy variables prevents us from having to make multiple models to represent different groups (Garavaglia and Sharma, 1998).

The annotated Python code containing all data preprocessing is included in the attached python script titled “D209\_T2.py”.

**C4. CLEANED DATA SET**

The final cleaned dataset is attached in the file “churn\_data\_processed\_T2.csv”.

**D1. SPLITTING THE DATA**

The “train\_test\_split” function from the sci kit learn library in python was used to split the data into training and test datasets. The test set was set to 30% of the original dataset, with the remaining 70% being used in the training dataset. All datasets were exported to csv files and are attached in the files “Xtrain\_T2.csv”, “Xtest\_T2.csv”, “ytrain\_T2.csv”, and “ytest\_T2.csv”.

**D2. OUTPUT AND INTERMEDIATE CALCULATIONS**

A screen shot of a computer program

Description automatically generated with low confidence After training the random forest with the training datasets, the model accuracy was assessed using the testing datasets. Model predictions were run using the test feature dataset (Xtest\_T2.csv). The resulting model predictions were then compared to the classification test dataset (ytest\_T2.csv). To compare model predictions to the test data, the sci-kit learn “metrics” module was used (Pedregosa et al., 2011). This module includes functions to calculate the MSE and R squared for the model. In addition to calculating the MSE and R squared using sci-kit learn, model errors were calculated and used to determine mean absolute percentage error (MAPE). MAPE was then used to estimate model accuracy. These calculations are shown below.

**D3. CODE EXECUTION**

The python script for all data preparation and analysis was attached (D209\_T2.py). The python script was written and executed in Visual Studio Code.

**E1. ACCURACY AND MSE**

A screenshot of the output of MSE, R squared, and accuracy is provided below.

A black background with white text

Description automatically generated with low confidence

The MSE of the model is 2.13, the R2 is 0.997, and the calculated accuracy is 88.99%. All of these metrics point to a good predictive model. A MSE and R2 close to 1 and model accuracy greater than 80% without feature elimination or hyperparameter tuning indicates that this model is strong as is. There are some simple things, like hyperparameter tuning and feature selection, that may improve the model. However, the initial model MSE, R2, and accuracy of the model all point to a usable model with good predictive ability as is.

**E2. RESULTS AND IMPLICATIONS**

The question addressed with this work was, “Can a random forest regression be used to predict telecom customer tenure?”. According to the results, we can create a random forest to predict customer tenure with 88.99% accuracy and an R2 of 0.997. This model could be used to identify customers likely to have low tenure, but the telecom company should know that the model predictions are not 100% accurate. Therefore, caution and understanding of the model error should be used if the model is implemented in its current state. The accuracy of this model implies that there may be some room for improvement the model that may result in better predictions of customer tenure.

**E3. LIMITATION**

One limitation of this data analysis is that random forest regressions cannot extrapolate predictions outside of the initial range of the training data. This means that our predictions are limited to the scope of the original training dataset. If more data is collected that fall out of this initial range, it will be important to retrain and update the model for accuracte predictions.

**E4. COURSE OF ACTION**

Based on the model outcome, the model has high accuracy and can be used to make some predictions as is. However, as we did not do any hyperparameter tuning or feature elimination, my first suggestion would be to attempt to improve the model through those processes. This may only result in slight improvement in model accuracy, as the model already has high accuracy, but taking these additional steps could make the model run faster and streamline predictions later. These slight improvements in model performance will be important if the company hopes to use the model on a larger scale.

**F. PANOPTO RECORDING**

The Panopto link was attached. The viewer link is:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f34524fd-bd3e-4c76-b659-b03000bf82e0>

**G. SOURCES FOR THIRD-PARTY CODE**

No sources of third-party code were used.

**H. SOURCES**

Breiman L (2001) Random forests. Machine learning, 45(1): 5-32.

Garavaglia, S., & Sharma, A. (1998). A smart guide to dummy variables: Four applications and a macro. In Proceedings of the northeast SAS users group conference (Vol. 43).

Harris, C.R., Millman, J.K., van der Walt, S.J, Gommers, R., Virtanen, P., Cournapeau, V., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernandez del Rio, F., Wiebe, M., Peterson, P., Gerard-Marchant, P., Sheppard, K., … Oliphant, T.E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362.

Lyashenko, V. (2021). Random Forest regression - the definitive guide. cnvrg. https://cnvrg.io/random-forest-regression/

McKinney, W. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.