Jordan Tompkins

D209, Data mining I task 2

A1, Proposal of Question: The question that was looked at for this analysis was, can we predict the “MonthlyCharge” for a customer? To answer this question, the Random Forest prediction method was used.

A2, Defined Goal: One goal for this analysis and question is to use the random forest prediction method, to create a model that can help the company predict how much a customer will be charged. The goal is to create a model that the company might be able to use to help them accurately predict a customers bill, so they can determine how and where to focus efforts on making the most money.

B1, Explanation of Prediction Method: Random forest is a machine learning algorithm that “…combines the output of multiple decision trees to reach a single results” (“What is Random Forest?”). Rather than just using one decision tree and using that result as the result for the predictive analysis, Random forest will use multiple decision trees to create their own predictions before it averages these predictions. This is then used as the result of the random forest prediction. For example, say you have 3 decision trees being used to predict the monthly charge for a customer. Each tree will produce their own results and predictions. Random forest will then take the average of these and that is the single result produced. So if the predictions of the three trees are $50, $100 and $75, the average of that will be $75 and that is the result that the random forest algorithm will produce.

B2, Summary of Method Assumption: The random forests prediction method has a few assumptions. One of the big assumptions for it is that there are no formal distributions. It is a non-parametric model that can handle skewed and multi-modal data (Vishalmendekarhere). This allows for the usage in data that isn’t necessarily linear. Rather than using a linear regression model, you can use a random forest which will allow you to better create a model for prediction with the non-linear data.

B3, Packages or Libraries List:

|  |  |
| --- | --- |
| Package and libraries | Usage |
| Pandas | This was used to import that data into the data frame and helped with any data preparation |
| SelectKBest, f\_classif from sklearn.feature\_selection | Using SelectKBest and f\_classif we were able to determine what the best features to use in our model would be. |
| Train\_test\_split from sklearn.model\_selection | Train\_test\_split allows us to breakup the data into training sets and testing sets so the whole dataset was not being used for the analysis. It allowed for us to train the model on one set, the training set, and then test it on a separate set, the testing set. |
| Mean\_squared\_error from sklearn.metrics | This was used to calculate the mean squared error, one of the testing metrics used for the Random Forest prediction |
| RandomForestRegressor from sklearn.ensemble | RandomForestRegressor is used to perform the analysis. |

C1, Data Preprocessing. One of the main data preprocessing goals related to Random Forest prediction was to make sure that the data was prepped and ready for usage. To do this, we needed to clean the data and make sure that there were no nulls or missing values. The data was checked for any missing or null values, and categorical variables were encoded from the original responses to numeric responses of 1s and 0s.

C2, Data Set variables: The variables used to perform the analysis include “Tenure” which is a continuous variable, “Yearly\_equip\_failure” and “Bandwidth\_GB\_Year” which are also continuous variables. The categorical variables used in the analysis include “Multiple\_Yes”, “InternetService\_Fiber Optic”, “InternetService\_None”, “Gender\_Nonebinary” and “Churn\_Yes”. These were the variables used in the prediction of “MonthlyCharge”, a continuous variable which we were predicting in the analysis.

C3, Steps for Analysis: After loading the dataset using “pd.read\_csv” we had to first ensure that there were no missing or N/A values in the data using “df.isnull().sum()”. After seeing that there were no nulls or N/As, the next step was to drop the initial variables that were deemed unnecessary for the performance of the analysis. To do this, we used “df.drop” and specified the columns that we wanted to initially drop. The majority of the columns dropped were customer demographic specific or survey responses. Those variables were deemed unnecessary so they were dropped. After dropping those columns, dummy variables were created on all the categorical variables in the data set using “pd.get\_dummies()”, setting drop\_first equal to “True”. The data was cleaned and prepped to use but in order to get the best variables for the analysis, the “SelectKBest” method was called on the data set which allowed for the data to then be used to determine which features would be best for Random Forest prediction. Checking for any features with p-values greater than 0.05, those features were selected for the final analysis. See jupyter notebook for a copy of the code described above.

C4, Cleaned Data Set: See copy of cleaned data set uploaded separately.

D1, Splitting the Data: After cleaning the data and ensuring that it is ready for analysis, the next step was to split the data into training and testing sets. Arrays were first created for the data, using “X = df[features\_to\_keep]” and ‘y = df[‘MonthlyCharge’]”. The X array was all of the predictor variables determined from the preprocessing steps using “SelectKBest”, while the y array was the target variable of “MonthlyCharge”. After creating the arrays, “train\_test\_split” was used, with the two arrays, a test size of 30% and a random seed set to 13 to ensure reproducibility. The test size was set to 30% which meant that we would test 30% of the data in the analysis, and that left 70% of the data to the train, or fit the model. See copies of training and testing sets in files uploaded separately.

D2, Output and Intermediate Calculations: To complete this predictive analysis, random forests were used. The analysis was used to predict how much a customer would be charged per month which the company would then collect for revenue. After all the data was cleaned and prepped for analysis, testing and training sets were created using the “train\_test\_split” function where the data was broken up into training and testing sets. After the data was split, “RandomForestRegressor” and instantiated for use. The random forest regressor was then used to fit the training data set, X\_train and y\_train, before being used for prediction and testing, using the testing data set created. After the model was trained and predictions were determined, the accuracy of the model and the mean squared error of the model were calculated. These scores help determine how well a model does in terms of predictions. See the calculations below:

Graphical user interface, text

Description automatically generated

D3, Code execution: See jupyter notebook uploaded separately for the execution of the code to perform the analysis.

E1, Accuracy and MSE: Two of the key metrics that were looked at for this analysis were the accuracy and mean squared error, or MSE. The accuracy represents the models R-Squared value. To get this score, the “.score()” method was used. The model created had an accuracy, or R-squared value of 0.71. The accuracy, or R-squared shows how much variance is attributed to the predictor variables. So in this case, 0.71 or 71% of the variance in the models target can be attributed to the predictor variables. The other metric, mean squared error or MSE, shows the difference between the predicted values and the actual values. This model’s MSE was 538.20. When looking at MSE, it is typically better to have a lower value. So the 538.20 is slightly on the higher range, but given the variables used and the R-squared of the model, it is not an unreasonable calculation.

E2, Results and Implications: Using a random forest algorithm to help predict how much a customer will be charged is very helpful when it comes to a business predicting how much they are going to be making from the business. Using this algorithm you are able to obtain different metrics such as the accuracy, or R-squared, and the mean squared error, or MSE. In the model created, the accuracy, or R-squared was 0.71. This shows us that the 71% of the variance in the models prediction is attributed to the predictor variables used. When it comes to determining if the model is a good model, looking at the r-squared can be helpful. In this case, 0.71 is a relatively good score. Another metric that you can look at is the mean squared error, or MSE. The MSE shows you the difference between the predicted values and the actual values. It shows you how close the estimates are to the actual values. In our model, the MSE was 538.20. Typically, the lower the MSE, the better, but the MSE in this model is relatively good. While the R-squared and MSE are both good indicators that the model created is a good model, they can always be improved. The company can use different variables, increase the number of variables, etc. in order to help them create a better model and raise the accuracy while lowering the MSE.

E3, Limitation: While random forest is a popular machine learning model used for predictive analysis, it does come with its limitations. One of those limitations is that it can’t extrapolate: “However, it is important to know you data and keep in mind that a Random Forest can’t extrapolate. It can only make a prediction that is an average of previously observed labels” (Thomspon, 2019). This can limit the analysis due to the fact that your predictions can only go as high, or as low as the training data. So if the training data set has a small range of data compared to the whole data set, your prediction using a Random Forest is going to be within that small range, essentially neglecting the larger range of the original data set.

E4, Course of Action: There is a lot that the company might be able to do with this information. With a R-squared of 0.71 and a MSE of 538.20, the company knows that they can use this model and have decent predictions. However, they could use hyperparameter tuning to create an even better model. They could select more variables, or even better variables that could help improve the R-squared and reduce the MSE. There are multiple different ways that the company can use this information to help them improve their ability to predict how much a customer is going to pay per month for their services. If the company just wanted to use the model created, they could do so with the knowledge that the model created is a good model to use.

F, Panopto Recording: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=92e80d4d-9599-4940-8acd-afe50159dd9e>

G, Sources for Third-Party Code: No third-party code utilized.

H, Sources:

Thompson, B. (2019, December 20). *A limitation of Random Forest Regression*. Medium. Retrieved April 15, 2023 from

<https://towardsdatascience.com/a-limitation-of-random-forest-regression-db8ed7419e9f>

Vishalmendekarhere. (2021, January 22). *It’s All About Assumptions, Pros & Cons*. The Startup. Retrieved April 15, 2023 from

<https://medium.com/swlh/its-all-about-assumptions-pros-cons-497783cfed2d>

*What is Random Forest*? IBM. (n.d.) Retrieved from

<https://www.ibm.com/topics/random-forest>

‌

‌