|  |
| --- |
| Western Governors University |
| Data Mining 1 |
| D209 Task 2 |

|  |
| --- |
| Allison Casey  7-20-2024 |

**Part I: Research Question**

**A1: PROPOSAL OF QUESTION**

Can tenure be predicted using a random forest regressor model?

**A2: DEFINED GOAL**

The primary goal will be to create a random forest model that can help to predict a customer’s tenure so that a company can use this data to help make decisions to retain their customers for as long as possible.

**Part II: Method Justification**

**B1: EXPLANATION OF PREDICTION**

A decision tree creates a sequence of binary options forming a tree as it branches out with the final prediction or classification at the end of each path. Random forests will create multiple decision tree models and combine those predictions into a single result. This averaging allows the model to be better than a singular decision tree and should help with accuracy and overfitting. An expected outcome of this would be, to a certain degree of accuracy, this model will allow tenure to be predicted

**B2: SUMMARY OF METHOD ASSUMPTION**

One assumption of the random forest prediction method is that the data is accurate and representative. If the data being used to train the model is skewed for one feature or another and this skew isn’t representative of reality, then the model would also be skewed for that feature and be inaccurate.

**B3: PACKAGES OR LIBRARIES LIST**

|  |  |
| --- | --- |
| import os,sys | Used to set the directory to work out of |
| import pandas as pd | Used to import the data |
| import numpy as np | Used for array handling and capabilites |
| import seaborn as sns | Used to help create a correlation heatmap visualization |
| from sklearn.model\_selection import train\_test\_split | For splitting the data into train and test sets |
| from sklearn.model\_selection import GridSearchCV | To help get the best hyperparameters |
| from sklearn.ensemble import RandomForestRegressor | To instantiate the Random Forest model |
| from sklearn.metrics import mean\_absolute\_error as MAE, mean\_squared\_error as MSE | To calculate the MAE and MSE for the model |

**Part III: Data Preparation**

**C1: DATA PREPROCESSING**

One of the pre-processing goals for the data was to check for missing or null values in the data set and treat any if found.

**C2: DATA SET VARIABLES**

|  |  |
| --- | --- |
| Population | Numeric |
| Outage\_sec\_perweek | Numeric |
| Email | Numeric |
| Contacts | Numeric |
| Yearly\_equip\_failure | Numeric |
| Tenure | Numeric |
| MonthlyCharge | Numeric |
| Bandwidth\_GB\_Year | Numeric |
| Port\_modem | Categorical |
| Tablet | Categorical |
| Phone | Categorical |
| OnlineSecurity | Categorical |
| OnlineBackup | Categorical |
| DeviceProtection | Categorical |
| TechSupport | Categorical |
| PaperlessBilling | Categorical |
| Contract | Categorical |
| InternetService | Categorical |
| PaymentMethod | Categorical |

**C3: STEPS FOR ANALYSIS**

The submission accurately explains each step used to prepare the data for analysis, and the submission identifies an accurate code segment for each step.

3. Explain the steps used to prepare the data for the analysis. Identify the code segment for each step.

1. Detect and treat missing values

*df.info()*

*#Detect and treat NULLS - we can see from the describe call that there is only one column with missing values*

*#InternetService: This column actually doesn't have any null values it just got read in as NULL rather than None indicating the customer doesn't have this service*

*df['InternetService'].fillna('None',inplace=True)*

1. Drop the initial columns that won’t be used (demographic data and survey questions)

*df = df.drop(columns=[‘Churn’,'Customer\_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'TimeZone', 'Area', 'Job', 'Children', 'Age', 'Income', 'Marital', 'Gender', 'Techie', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'])*

1. Review summary statistics for outliers or other concerns

*df.describe()*

*categorical\_columns = ['Churn','Port\_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'Contract', 'InternetService', 'PaymentMethod']*

*for x in categorical\_columns:*

*print(df[x].describe())*

1. One hot encode the categorical variables and rename the columns for clarity

*#One hot encoding*

*df = pd.get\_dummies(df, columns=categorical\_columns, drop\_first=True, dtype = int)*

*#Rename the columns for clarity and to remove spaces*

*df = df.rename(columns = {'Contract\_One year':'Contract\_One\_Year', 'Contract\_Two Year':'Contract\_Two\_Year','InternetService\_Fiber Optic':'InternetService\_Fiber\_Optic', 'InternetService\_Fiber Optic':'InternetService\_Fiber\_Optic', 'PaymentMethod\_Credit Card (automatic)':'PaymentMethod\_CC', 'PaymentMethod\_Electronic Check':'PaymentMethod\_ECheck', 'PaymentMethod\_Mailed Check':'PaymentMethod\_MCheck'})*

1. Use a heatmap showing correlation to remove highly correlated features

*#Look for correlation in the data and remove highly correlated features*

*sns.heatmap(data = df[columns].corr(), annot=False, cmap='coolwarm')*

*A screen shot of a graph

Description automatically generated*

*#Remove the higher correlated features based on the heatmap*

*df = df.drop(columns = ['Multiple\_Yes', 'StreamingTV\_Yes', 'StreamingMovies\_Yes'])*

1. Export the prepared data

*df.to\_csv('PREPARED\_churn\_clean\_data.csv')*

See D209\_Task2.ipynb for the full code

**C4: CLEANED DATA SET**

PREPARED\_churn\_clean\_data.csv

**Part IV: Analysis**

**D1: SPLITTING THE DATA**

The submission provides reasonably proportioned training and test data sets.  
1. Split the data into training and test data sets and provide the file(s).

Data split and with a test size of .2:

*# Set up the data as X and y and split the data for training and testing*

*X = df.drop('Tenure', axis=1)y = df['Tenure']*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)*

*# Export training and test datasets*

*np.savetxt('X\_train.csv', X\_train, delimiter=',')*

*np.savetxt('X\_test.csv', X\_test, delimiter=',')*

*np.savetxt('y\_train.csv', y\_train, delimiter=',')*

*np.savetxt('y\_test.csv', y\_test, delimiter=',')*

See the following files for the exported split data:

X\_train.csv

X\_test.csv

y\_train.csv

y\_test.csv

see D209\_Task2.ipynb for full code

**D2: OUTPUT AND INTERMEDIATE CALCULATIONS**

The submission accurately describes the analysis technique used to appropriately analyze the data, and the submission includes accurate screenshots of the intermediate calculations performed.

2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.  
The analysis technique that was used on the data was a random forest regressor which is an ensemble learning method that constructs multiple decision trees and provides a combined output of the predictions. First the data was spit as explained above and then GridSearchCV was used to perform hyperparameter tuning to help identify the best parameters to set for the model.

A screen shot of a computer program

Description automatically generated

These parameters were then used in building the final model

**A screen shot of a computer screen

Description automatically generated**

The model was then fit to the training data set and used to predict with the test data so that it could be evaluated for r-squared, mean absolute error, mean squared error, and root mean squared error.

A screen shot of a computer program

Description automatically generated

**D3: CODE EXECUTION**

D209\_Task2.ipynb

**Part V: Data Summary and Implications**

**E1: ACCURACY AND MSE**

The r-squared is a value from 0 to 1 that indicates how well the regression model explains the observed data where higher numbers are better. The model had a good r-squared value.

* r-squared: .977 or 97.7%

The mean absolute error indicates the absolute error between the actual values (y\_test) and the predicted values (y\_pred) where 0 would mean a perfect prediction. The model had a pretty good MAE score.

* MAE: 3.255

The mean squared error measures how close a regression line is to a set of data points so the less the MSE the small the error and the better the estimator is. For this model the MSE was relatively close to zero so it is an acceptable score.

* MSE: 15.85

The root mean squared error is an extension of the MSE and lower values also indicate a better model. For this model the RMSE was also close to zero making it an acceptable value.

* RMSE: 3.98

Based on the evaluation metrics the model is fairly accurate.

**E2: RESULTS AND IMPLICATIONS**

The model scores decently for all the metrics that were calculated. The r-squared value is very close to 1 and the MAE, MSE, and RMSE are all relatively close to zero. The evaluation metrics show that the model could be useful to predict the tenure of customers.

**E3: LIMITATION**

One limitation of the model is that it is very reliant on the training and testing data being accurate and representative. If the data used to build the model is not good, then the model is not good. Having more data to build and test the model with could be beneficial.

**E4:** **COURSE OF ACTION**

Based on the model evaluation metrics, this random forest model could be used to help predict customer tenure. However, it would also be advisable to try to continue to improve the model and data. Gathering more data and having more data to test and train the model against could help make it better. The model could likely also be improved by doing more hyper-parameter tuning on parameters that weren’t tuned in this model. It would also be smart to use the model to dig deeper into exploring which features contribute to longer tenure as well as potentially building more models that consider the demographic data that was left out of this model.

**Part VI: Demonstration**

**F: PANOPTO RECORDING**

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=2a00e8a4-f2de-496f-9637-b1b3016cf74f

**Sources**

Ahmed, M Waqar. “Understanding Mean Absolute Error (MAE) in Regression: A Practical Guide.” *Medium*, Medium, 24 Aug. 2023, medium.com/@m.waqar.ahmed/understanding-mean-absolute-error-mae-in-regression-a-practical-guide-26e80ebb97df.

Elleh, Fes. “Task 2: Predictive Analysis.” D209 Data Mining 1. 2024.

Frost, Jim. “Root Mean Square Error (RMSE).” *Statistics By Jim*, 28 May 2023, statisticsbyjim.com/regression/root-mean-square-error-rmse/.

“Randomforestregressor.” *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. Accessed 20 July 2024.