D209 Data Mining 1 Performance Assessment

Task 2: Predictive Analysis

Author: Eric Colwell

Western Governors University

This report deals with the rubric for Western Governors University course D209 Data Mining 1 Task 2 and answers all the items in rubric order.

# **Part I:**

**A1:**

The research question for this analysis is “Using the Random Forest Regression method, is it possible to predict the tenure of a customer using the data available in the churn dataset?

**A2:**

The objective of this data analysis is to create a machine learning model using Random Forests to predict customers tenure with data from the churn dataset. For this analysis I have chosen Tenure as the dependent or target variable. For the initial model, the independent or predictor variables are Population, Area, Children, Age, Income, Marital, Gender, Churn, Outage\_sec\_perweek, Email, Contacts, Yearly\_equip\_failure, Techie, Contract, Port\_modem, Tablet, InternetService, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, PaperlessBilling, PaymentMethod, MonthlyCharge, and Bandwidth\_GB\_Year. Variables not included in the initial model are CaseOrder, Customer\_id, Interaction(UID), City, State, County, Zip, Lat, Lng, TimeZone, and Job. These variables have high cardinalities and would not add predictive power to the model. Also, the variables Item1 through Item8 are not included.

# **Part II:**

**B1:**

For this analysis I have chosen to use Random Forest Regression. Random Forest is a supervised machine learning method based on multiple Decision Trees and the ensemble learning method. The ensemble learning method in general is training multiple machine learning algorithms then combining their predictions in some way to make more accurate predictions. (Lyashenko). The expected outcome is that the Random Forest model can, with a high accuracy, predict customers tenure from the other calculated data points. Below is a visualization of a random forest.

Diagram

Description automatically generated(Chakure, 2022)

**B2:**

One assumption of Random Forest models is that sampling is representative. This is when data in the sample reflects the properties of the population. These samples can be used to generalize results to a population and can approximately predict the populations properties.

**B3:**

The benefits of using the Python language for this analysis are that it is a capable tool for data science, statistics, data exploration, data analysis, and predictive analytics. Also, I am more familiar with Python than R. For these reasons I chose this programming language for the Random Forest analysis. I imported pandas to import the dataset, data manipulation, and analysis. I imported numpy for numerical calculations. I imported matplotlib.pyplot for plotting data. I imported missingno to visually check for missing values. I imported train\_test\_split for data splitting and model testing. I imported variance\_inflation\_factor from statsmodels to check VIF scores. I imported python\_version to show the version I am working with. I imported SelectKBest for variable reduction. I imported RandomForestRegressor for model creation. I imported GridSearchCV for model hyperparameter tuning. I imported make\_scorer as a scoring function. I imported mean\_squared\_error for model analysis. I imported r2\_score for model analysis. I imported mean\_absolute\_error for model analysis. I imported accuracy\_score for model analysis. I imported warnings to ignore the warnings.

# **Part III:**

**C1:**

My data preparation goals follow the Data Analytics Life Cycle. One important preprocessing goal is to encode the categorical variables. For Random Forest models this step is crucial. Random Forest models will not work with categorical data. Due to this limitation, I used ordinal encoding for the categorical variables with Yes/No, and one-hot-encoding for categorical variables with cardinality higher than 2.

**C2:**

Below are the summary statistics for the target variable and all predictor variables including screenshots of output showing the count of instances (10,000), mean (average), standard deviation (amount of variation of a set of data), minimum (least amount), quartile ranges (measure of variability of the data), maximum (largest amount) for continuous variables. Unique values, and value counts for categorical variables.

**Tenure (Target Variable):** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the number of months the customer has stayed with the provider (range roughly from 1 to 72).

Table

Description automatically generated with medium confidence

**Population:** Datatype is int64, quantitative, continuous, and has no null values. The value reflects the population within a mile radius of the customer, based on census data (range from 0 to 111,850).

Table

Description automatically generated

**Area:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the area type (Rural, Urban, Suburban), based on census data.

Table

Description automatically generated

**Children:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the number of children in the customer’s household as reported in sign-up information (ranging from 0 to 10).

Table

Description automatically generated

**Age:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the age of the customer as reported in sign-up information (ranging from 18 to 89).

Table

Description automatically generated

**Income:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the annual income of the customer (or invoiced person) as reported at time of sign-up (ranging from roughly 348 to 258,900).

Text

Description automatically generated

**Marital:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the marital status of the customer as reported in sign-up information (Divorced, Married, Never Married, Separated, Widowed).

Text

Description automatically generated

**Gender:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the customers self-identification as male, female, or nonbinary.

Text

Description automatically generated

**Churn:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer discontinued service within the last month (Yes, No).

Graphical user interface, text

Description automatically generated

**Outage\_sec\_perweek:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the average number of seconds per week of system outages in the customer’s neighborhood (range from roughly 0.1 to 21.2).

Text

Description automatically generated

**Email:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the number of emails sent to the customer in the last year (range from 1 to 23).

Table

Description automatically generated with medium confidence

**Contacts:** Datatype is float64, quantitative, discrete, and has no null values. The value reflects the number of times the customer contacted technical support (range from 0 to 7).

Table

Description automatically generated

**Yearly\_equip\_failure:** Datatype is float64, quantitative, discrete, and has no null values. The value reflects the number of times customers equipment failed and had to be reset/replaced in the past year (range from 0 to 6).

Table

Description automatically generated

**Techie:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer considers themselves technically inclined (Yes, No).

Text

Description automatically generated

**Contract:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the contract term of the customer (Month-to-month, One Year, Two Year).

Text

Description automatically generated

**Port\_modem:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has a portable modem (Yes, No).

**Text

Description automatically generated**

**Tablet:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer owns a tablet such as iPad, Surface, etc. (Yes, No).

**Graphical user interface, text

Description automatically generated**

**InternetService:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the customers internet service provider (DSL, Fiber Optic, or None).

**Text

Description automatically generated**

**Phone:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has a phone service (Yes, No).

**Text

Description automatically generated**

**Multiple:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has multiple lines (Yes, No).

Text

Description automatically generated

**OnlineSecurity:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has an online security add-on (Yes, No).

**Text

Description automatically generated**

**DeviceProtection:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has device protection add-on (Yes, No).

**Text

Description automatically generated**

**TechSupport:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has a technical support add-on (Yes, No)

**Text

Description automatically generated**

**StreamingTV:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has streaming TV (Yes, No).

**Text

Description automatically generated**

**StreamingMovies:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has streaming movies (Yes, No).

**Graphical user interface, text

Description automatically generated**

**PaperlessBilling:** Datatype is object, qualitative, categorical, and has no null values. The value reflects whether the customer has paperless billing (Yes, No).

Text

Description automatically generated

**PaymentMethod:** Datatype is object, qualitative, categorical, and has no null values. The value reflects the customer’s payment Method (Bank Transfer, Credit Card, Electronic Check, Mailed Check).

Text

Description automatically generated

**MonthlyCharge:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the amount charged to the customer monthly. This is an average for the customer. For brand new customers, this value is the average for other customers who fit the new customers profile (range from roughly 80 to 290).

Text

Description automatically generated

**Bandwidth\_GB\_Year:** Datatype is float64, quantitative, continuous, and has no null values. The value reflects the average amount of data used, in GB, in a year by the customer (range from roughly 155 to 7159).

Text

Description automatically generated

**C3:**

I imported the churn.csv file using pandas.

Text

Description automatically generated

Next, I checked the dataset shape.

Graphical user interface, text, application

Description automatically generated

There are zero null values, 10,000 rows, and 50 columns.

A picture containing chart

Description automatically generated

I checked for duplicated rows and found there were none.

Graphical user interface, text, application

Description automatically generated

I made a copy of the original to preserve it.

Text

Description automatically generated

I made a new dataset that included the variables or columns that I used for the analysis.

Text, letter

Description automatically generated

I checked the uniqueness of each of those chosen variables for validity. I did this for all variables included in the initial model but am just showing one code example.

Graphical user interface, text, application

Description automatically generated

I made a new dataframe (df2) for a Random Forest model and includes only the variables being utilized in the initial model.

Text

Description automatically generated

For the data wrangling phase I used ordinal encoding for the variables with Yes/No values and set them to 0 for No and 1 for Yes.

Text, letter

Description automatically generated

Then I used pd.get\_dummies with k-1 to encode the nominal variables.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The dataset is now clean and prepared for initial Random Forest model. There are more steps that I took to reduce the number of variables in the model. They are shown in part D1.

**C4:**

The Jupyter notebook file (.ipynb format) containing the complete annotated code along with a copy of the cleaned dataset named ‘Eric\_Colwell\_clean\_dataset\_task2’ in .csv format for this project will be uploaded with the submission.

# **Part IV:**

**D1:**

I used the SelectKBest method to filter out variables with p-values greater than 0.05 and reduced the initial model from 38 predictor variables down to 6.

Graphical user interface, text, application

Description automatically generated

Then I checked for high correlation between variables. Bandwidth\_GB\_Year had a high correlation to Tenure at 0.99, so I removed it.

Table

Description automatically generated

Next, I check the variance inflation factors for the predictor variables that remained. There were no VIF scores greater than 10 so none were removed.

Text

Description automatically generated

Next, I split the data into 70% training and 30% testing datasets. These files will be uploaded along with the submission as .csv files named Eric\_Colwell\_task2\_X\_train\_dataset, Eric\_Colwell\_task2\_X\_test\_dataset, Eric\_Colwell\_task2\_y\_train\_dataset, and Eric\_Colwell\_task2\_y\_test\_dataset. The code snippet for splitting the data is shown below.

Text

Description automatically generated

**D2:**

I instantiated the RandomForestRegressor, then I set the parameters for the GridSearchCV.

Text

Description automatically generated

Next, I ran the GridSearchCV to find the best set of parameters.

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

I decided to do another GridSearchCV with additional parameters.

Text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

For analysis of the best model, I used MSE (mean squared error), RMSE (root mean squared error, and r2 (r-squared). I checked both the training and testing data.

Text

Description automatically generated

Text

Description automatically generated

Next, I checked the importance of the features used in the model.

Graphical user interface, chart

Description automatically generated

**D3:**

Code snippets used to perform the classification analysis were shown in the screenshots above. I will upload a .ipynb Jupyter notebook with the complete code for this project along with the rest of the submission.

**Part V:**

**E1:**

The MSE of the training data is 0.24 and the RMSE is 0.49. The MSE of the testing data is 543.32 and the RMSE is 23.31. The mean of the difference between the predicted and actual data points is squared to remove the impact of positive or negative signs. This is called MSE or mean squared error. The square root of the mean squared error shows an average of how much the predicted values are off. This is called RMSE or root mean squared error. For this analysis the predictions for tenure are off on average of 23.31 months.

**E2:**

The Random Forest model that I created had a RMSE of 23.31 on the testing data. This shows that the model performed very poorly. The mean of Tenure is 34.53 with a minimum of 1 and a maximum of 72. If I simply predicted the mean of 34.53 it wouldn’t be much worse than what the models predictions are. Due to this implication, the results of the model must be improved upon to be useful in making predictions.

**E3:**

Random Forest is a simple but effective machine learning model that is widely used. Its ability to predict data that is non-linear makes it useable for many datasets. One limitation is that it cannot extrapolate data. The predictions it makes are an average of previous data points. Covariate shift is when training and testing inputs differ in their range. When this happens, Random Forest does not perform well because it cannot extrapolate.

**E4:**

Based on the output from this analysis the company should not use this model to predict the tenure of customers. I would recommend further analysis with different feature reduction methods, scaling, and removal of outliers. These steps could cause the model to perform more accurately and lead to a model that could be used successfully by the company to predict customers tenure.

**Part V:**

**F:**

The Panopto video recording URL will be uploaded as part of the submission.

References

Chakure, A. (March 07, 2022). Retrieved from <https://builtin.com/data-science/random-forest-python>

Lyashenko, V. (n.d.). Retrieved from <https://cnvrg.io/random-forest-regression/>