**CAN THE TELECOM FIRM FORECAST WHETHER A CUSTOMER WILL CHURN BASED ON CIRCUMSTANCES AND DATA OBTAINED IN THE RELIABILITY?**

**NVM2 — NVM2 TASK 2: PREDICTIVE ANALYSIS**

**DATA MINING I — D209**

**PRFA — NVM2**

**STUDENT NAME: IBRAHIM SULEIMAN**

**DATA ANALYTICS**

**STUDENT NUMBER: 001429984**

**COLLEGE OF INFORMATION TECHNOLOGY, WESTERN GOVERNORS UNIVERSITY**

**NAME OF INSTRUCTOR: FESTUS ELLEH**

**SEPTEMBER 11th, 2022**

**Part I: Research Question**

1. **Describe the purpose of this data mining report by doing the following:**

**1.  Propose one question relevant to a real-world organizational situation that you will answer using one of the following prediction methods:**

**•  decision trees**

**•  random forests**

**•  advanced regression (i.e., lasso or ridge regression).**

Can the telecom firm forecast whether a customer will churn based on circumstances and data obtained in the reliability? That would be an important business decision for the telecom company and its leadership. Random forests will be the technique used to find the answer to this query.

**2.  Define one goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.**

Finding out if it is possible to anticipate a customer's churning based on both categorical and continuous factors is one aim of this data study.

**Part II: Method Justification**

**B.  Explain the reasons for your chosen prediction method from part A1 by doing the following:**

**1.  Explain how the prediction method you chose analyzes the selected data set. Include expected outcomes.**

Using training and test datasets produced from a primary dataset, the Random Forest analysis approach examines data. Tony Yiu (2019) claims that it makes use of ensemble learning, which combines classifiers to offer solutions.

It makes use of numerous decision trees, each of which is constructed using feature randomness. This is an effort to make a prediction that is more accurate than any one individual tree by using an uncorrelated forest of trees acting as a committee.

**2.  Summarize one assumption of the chosen prediction method.**

In terms of distribution, Random Forest makes no formal assumptions. According to Sarah Richmond (2016), it is non-parametric and supports ordinal and non-ordinal categorical data, as well as skewed and multi-modal data.

**3.  List the packages or libraries you have chosen for Python or R, and justify how *each* item on the list supports the analysis.**

* Pandas are a common import for projects in machine learning. It offers tools for parsing and scoring data, in addition to ways of accessing and displaying data.

- Numpy is a popular import for applications in machine learning and an approach can improve reading and visualizing data, in addition to statistical tools for information processing as well as evaluation.

* The standard visualization import is Matplotlib. The capabilities for visualizing reports and data points in this package are more powerful.
* Graphs, charts, and matrices from Seaborn are illustrative and intuitive to the eye.

- Scikit-learn offers strategies and justifications for dividing, training, testing, and fitting data.

Additionally, this package includes justifications for categorizing and forecasting data as well as applying metrics to models (Michael Galarnyk,2018).

**Part III: Data Preparation**

**C.  Perform data preparation for the chosen data set by doing the following:**

**1.  Describe one data preprocessing goal relevant to the prediction method from part A1.**

Scaling the variables used for the study would be one data preprocessing objective relevant to the Random Forest approach. This will guarantee that they are weighed equally and contribute to more precise forecasts.

**2.  Identify the initial data set variables that you will use to perform the analysis for the prediction question from part A1, and group *each* variable as continuous or categorical.**

The desired variable, churn, is categorical. Continuous predictor variables include yearly equip failure, age, dependability, children's income, indications of active listening, and reliability. Two categorical predictor variables are "Contract" and "Techie."

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

Before beginning the analysis, the data must be prepared. Making sure there are no blank columns in any of the columns is the first step. The data in the columns will then need to be checked to make sure there are no duplicates. Additionally, we'll want to verify that there are no duplicate columns or rows, so we'll check that and make sure the outcome is (False).

The dataset contains a number of columns that were determined to be unrelated to the predictive analysis, such as customer demographics that are linked to interactions and the location of the consumer and cannot be modified. These columns should be removed from the dataset. This facilitates working with the data. It will be necessary to change the category variables to (1 or 0), correspondingly. The data must also be divided into training and test sets, and the predictor variables must be scaled.

**4.  Provide a copy of the cleaned data set.**

|  |
| --- |
| **import numpy as np**  **import pandas as pd**  **from sklearn import linear\_model**  **import matplotlib.pyplot as plt**  **import seaborn as sns**  **%matplotlib inline**  **import sklearn**  **import scipy**  **import statsmodels.api as sm**  **from scipy import stats**  **import matplotlib.pyplot as plt**  **from sklearn import datasets**  **from sklearn import preprocessing**  **from sklearn.ensemble import RandomForestClassifier**  **from sklearn.model\_selection import train\_test\_split**  **from sklearn.metrics import accuracy\_score**  **from sklearn.model\_selection import cross\_val\_score, train\_test\_split**  **from sklearn import metrics**  **from sklearn.metrics import classification\_report**  **from sklearn.preprocessing import StandardScaler**  **pd.set\_option('display.max\_columns', None)**  **print("pandas version " + pd.\_\_version\_\_)**  **print("numpy version " + np.\_\_version\_\_)**  **print("scipy version " + scipy.\_\_version\_\_)**  **print("seaborn version " + sns.\_\_version\_\_)**  **print("statsmodels version " + sm.\_\_version\_\_)**  **pip install scikit-plot**  **# Ignore Warning Code**  **import warnings**  **warnings.filterwarnings('ignore')**  **df = pd.read\_csv('churn\_clean.csv')**  **df.head()**  **df.info()**  **#check for missing data**  **df.isna().any()**  **#check for duplicate data in columns**  **df[df.duplicated()]**  **# check if any cols are duplicated - Looking for False**  **df.columns.duplicated().any()**  **# drop demographic data**  **df = df.drop(['CaseOrder','Customer\_id','Interaction','UID','City','State','County','Zip','Lat','Lng','Population','Area','TimeZone','Job'], axis=1)**  **# verify columns were dropped**  **df.head()**  **#view descriptive statistics**  **df.describe()**  **df.rename(columns={'Item1':'Timely response','Item2':'Timely fixes','Item3':'Timely replacements','Item4':'Reliability','Item5':'Options','Item6':'Respectful response','Item7':'Courteous exchange','Item8':'Evidence of active listening'},inplace=True)**  **df.head()**  **newdf=df[['Age','Reliability','Children','Income','Evidence of active listening','Yearly\_equip\_failure','Contract','Techie','Churn']].copy()**  **newdf**  **#change yes/no to 1/0**  **newdf = newdf.replace(to\_replace = ['Yes','No'],value = [1,0])**  **#change Contract, Techie and Churn columns to numeric**  **newdf['Contract'].replace(('One year','Month-to-month','Two Year'), (0,1,2), inplace=True)**  **newdf**  **#export prepared dataset**  **newdf.to\_csv('prepared\_d209task2.csv', index = False)** |

**Part IV: Analysis**

**D.  Perform the data analysis and report on the results by doing the following:**

**1.  Split the data into training and test data sets and provide the file(s).**

To make sure that the variables are measured fairly on the same scale, the data is scaled. The remaining 80% of the data were used for training, while the remaining 20% were used to build a training and testing set of data. The random forest classifier is then used with the data, with the proper parameters being set (see code for parameters). The task contribution includes copies of the training and test datasets.

**2.  Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.**

First, the data in the new dataframe (the variables for analysis) must be scaled to ensure that they are all assessed equally because the values of the variables and the observations for the variables vary. The categorical predictor variable (text) must be converted from text to numerical form after the variables have been scaled, with the Yes and No values being changed to 1 and 0, respectively. Listed below are the scaled data from the dataframe:

Table

Description automatically generated

To enable the algorithm to make predictions based on the training data, the scaled data must next be transformed into the train and test datasets. The classification's parameters are set by the Random Forest Classifier, which is then launched.

In this model, n\_number of estimators is fixed to 100. Following that, the training set of data is fitted to this classifier. After that, the algorithm can generate predictions and compare them to the actual data. The model's accuracy, precision, recall, and F1-score can then be displayed in a classification report. These can be observed in the following example:

Table

Description automatically generated

Table

Description automatically generated

The true positive rate in comparison to the false positive rate can be shown by visualizing the ROC Curve. The model is better at predicting outcomes when the curve is higher. As can be observed from the graph below, the model has somewhat higher values on the ROC curve than normal, demonstrating a very high level of predictive power for true positive values.

**Chart, line chart

Description automatically generated**

**3.  Provide the code used to perform the prediction analysis from part D2.**

|  |
| --- |
| **#scale data**  **predictors = newdf.columns[newdf.dtypes.apply(lambda c: np.issubdtype(c, np.number))]**  **scaler=StandardScaler()**  **newdf[predictors] = scaler.fit\_transform(newdf[predictors])**  **#convert target variable to numeric**  **newdf['Churn']=df.Churn.map(dict(Yes=1, No=0))**  **#show scaled data**  **newdf.head()**  **train , test = train\_test\_split(newdf,test\_size=0.20, random\_state=42)**  **x\_train=train.drop('Churn',axis=1)**  **y\_train=train['Churn']**  **x\_test=test.drop('Churn',axis=1)**  **y\_test=test['Churn']**  **#export test and train files**  **x\_train.to\_csv(r'C:\Users\Ibrahim\Desktop\churn\_Xtrain-Task2.csv',index=False)**  **x\_test.to\_csv(r'C:\Users\Ibrahim\Desktop\churn\_Xtest-Task2.csv',index=False)**  **y\_train.to\_csv(r'C:\Users\Ibrahim\Desktop\churn\_Ytrain-Task2.csv',index=False)**  **y\_test.to\_csv(r'C:\Users\Ibrahim\Desktop\churn\_Ytest-Task2.csv',index=False)**  **from sklearn.ensemble import RandomForestClassifier**  **clf=RandomForestClassifier (bootstrap=True, class\_weight=None, criterion='gini',**  **max\_depth=None, max\_features='sqrt', max\_leaf\_nodes=None,**  **min\_impurity\_decrease=0.0,**  **min\_samples\_leaf=1, min\_samples\_split=2,**  **min\_weight\_fraction\_leaf=0.0, n\_estimators=100, n\_jobs=1,**  **oob\_score=False, random\_state=None, verbose=0,**  **warm\_start=False)**  **clf.fit(x\_train,y\_train)**  **#Saving predictions**  **y\_pred=clf.predict(x\_test)**  **#Showing predictions vs actual**  **pd.DataFrame(data={'Predicted': y\_pred, 'Actual': y\_test}).head(15)**  **#Showing predictions vs actual**  **pd.DataFrame(data={'Predicted': y\_pred, 'Actual': y\_test}).head(15)**  **#Classification report**  **from sklearn.metrics import classification\_report**  **print(classification\_report(y\_test, y\_pred))**  **#Show accuracy score**  **from sklearn.metrics import accuracy\_score**  **accuracy = accuracy\_score(y\_test, y\_pred)**  **print(accuracy)**  **#Show recall Score**  **from sklearn.metrics import recall\_score**  **recall\_score(y\_test, y\_pred, average='weighted')**  **#Show precision Score**  **from sklearn.metrics import precision\_score**  **precision\_score(y\_test, y\_pred, average='weighted')**  **#Show F1 Score**  **from sklearn.metrics import f1\_score**  **f1\_score(y\_test, y\_pred, average='weighted')**  **#Visualize ROC Curve**  **import matplotlib.pyplot as plt**  **import scikitplot as skplt**  **y\_probas=clf.predict\_proba(x\_test)**  **skplt.metrics.plot\_roc(y\_test, y\_probas, figsize=(10, 8))**  **plt.show()**  **#Determine AUC**  **from sklearn import preprocessing**  **from sklearn.metrics import roc\_auc\_score**  **# probs = y\_probas[:, 1]**  **# print ('ROC AUC =', roc\_auc\_score(y\_test, probs, multi\_class='ovo'))**  **def multiclass\_roc\_auc\_score(y\_test, y\_pred, average="macro"):**  **lb = preprocessing.LabelBinarizer()**  **lb.fit(y\_test)**  **y\_test = lb.transform(y\_test)**  **y\_pred = lb.transform(y\_pred)**  **return roc\_auc\_score(y\_test, y\_pred, average=average)**  **#Area Under Curve**  **from sklearn import preprocessing**  **from sklearn.metrics import roc\_auc\_score**  **roc\_auc\_score(y\_test, y\_pred)**  **#Mean Squared Error**  **from sklearn.metrics import mean\_squared\_error**  **mean\_squared\_error(y\_test,y\_pred)** |

**Part V: Data Summary and Implications**

**E.  Summarize your data analysis by doing the following:**

**1.  Explain the accuracy and the mean squared error (MSE) of your prediction model.**

The model's accuracy score is 0.708, which indicates that based on the detected variables, it has a 70.8% chance of properly predicting whether a customer will churn. This model's MSE is 0.292, which indicates a false positive prediction rate or error of 29.2%.

These data demonstrate how well the model predicts whether a customer will be churned depending on the variables when utilizing the variables that have been discovered.

**2.  Discuss the results and implications of your prediction analysis.**

The outcomes are determined by the categorization report values and the Area Under Curve (AUC) values. The findings indicate that the current model has a favorable prediction rate of 56-74%. These have the following values: 0.708 for recall, 0.6452 for precision 0.5277 for AUC.

The MSE is 0.292, or 29.2%, indicating an extremely low false positive rate (fewer errors in predicting outcomes). The model that uses the aforementioned characteristics to forecast whether a consumer will leave is extremely accurate, precise, and has a small error rate.

This model may be used to make a reliable prediction for the telecom company, but since it is not perfect, results may still require some human interaction to be confirmed. This would be negligible given the high accuracy level, as a 70.8% accuracy rate is frequently regarded as slightly above average.

**3.  Discuss one limitation of your data analysis.**

This study's usage of a single model constitutes one of its limitations (the Random Forest Classifier). Due to the number of analysis rounds, this is thought to be more robust than a standard decision tree, yet there was only one analysis performed. The number of estimators employed would also be a restriction.

The accuracy and precision of the model would be affected by a greater or lower value. And last, one limitation can be the variables employed. Other variables, with fewer or more options, could have been selected, leading to different ratings for those models, as they were by this analyst.

**4.  Recommend a course of action for the real-world organizational situation from part A1 based on your results and implications discussed in part E2.**

It is suggested that the telecommunications business adopt this model as a reliable foundation for estimating the likelihood that a client will leave. The factors and results would provide a solid indication of the likelihood of churning when compared to target consumer, allowing the experts to create a long-term customer relationship plan that might reduce the likelihood. The telecom business might use this to get prepared for churning.

Additionally, customer service agents and management can address the factors in question differently and more effectively by using them as markers of possible churn, which will lower that possibility. Last but not least, further research should be done to confirm that the variables found are the strongest indications and that the existing model is applicable.

**Part VI: Demonstration**

**F.  Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.**

You can view the session using the following link:  
<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=dcf42696-07f4-4081-b411-af120163d507>

**G.  Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable.**

**Web sources used**

*Grant, P. (2019). Understanding Multiple Regression; The fundamental basis behind this commonly used algorithm.*

Medium***.***[***https://towardsdatascience.com/understanding-multiple-regression-249b16bde83e***](https://towardsdatascience.com/understanding-multiple-regression-249b16bde83e)

Deepanshu, B. (2020). *How to Import Data in Python.*

RSGB Business Consultant Pvt. Ltd. <https://www.listendata.com/2017/02/import-data-in-python.html#Import-CSV-files>

Pierre-Louis B. (2020). *Principle Components Analysis(PCA), Fundamentals, Benefits & Insights for Industry.*

Medium. <https://towardsdatascience.com/principal-components-analysis-pca-fundamentals-benefits-insights-for-industry-2f03ad18c4d7>

John S. (2018). *Data Cleaning with python and Pandas: Detecting Missing Values.*

Medium. <https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b>

Angelica Lo D. (2021). *How to detect outliers with Python.*

Medium. <https://towardsdatascience.com/how-to-detect-outliers-with-python-pyod-aa7147359e4b>

Michael G. (2018). *Understanding Boxplots.*

Medium. <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51>

John S. (2018). *Data Cleaning with python and Pandas: Detecting Missing Values.*

Medium. <https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b>

**Reference**

Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. ISBN-13: 978-1-119-52684-1.

Yiu, T. (2021, September 29). Understanding random forest. Medium. Retrieved March 1, 2022,

from <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

Richmond, S. (2016, March 21). Algorithms exposed: Random Forest. BCCVL. Retrieved March 1, 2022,

from <https://bccvl.org.au/algorithms-exposedrandomforest/#:~:text=ASSUMPTIONS,are%20ordinal%20or%20non%2Dordinal>