**WHAT CUSTOMERS ARE MOST INCLINED TO CHURN AND WHAT FACTORS ARE MOST LIKELY TO MAKE A CUSTOMER CHURN?**

**NVM2 TASK 1 PA: CLASSIFICATION ANALYSIS**

**DATA MINING I — D209**

**PRFA — NVM**

**STUDENT NAME: IBRAHIM SULEIMAN**

**DATA ANALYTICS**

**STUDENT NUMBER: 001429984**

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

**NAME OF INSTRUCTOR: FESTUS ELLEH**

**SEPTEMBER 10th, 2022**

**Part I: Research Question A. 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 classification methods:**

What customers are most inclined to leave and what factors are most likely to make a customer leave? is a real-world business question that a classification study could assist in resolving. We will respond to this query by employing the K-Nearest Neighbor algorithm.

**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.**

The data on which consumers are likely to leave the telecommunications firm is to be sent to its management as a result of the data analysis. The research will offer accurate measurements to ensure that the organization has credibility in the data and will also provide evaluations for aspects that the telecommunications company should watch to lower the possibility of churn.

**Part II: Method Justification B. Explain the reasons for your chosen classification method from part A1 by doing the following:**

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

By using "k" nearest labeled data points as a voting block, a K-Nearest Neighbor forecasts the label of a data point. The method will look at the specified number of data points around the unclear data point to determine its label when the user-specified value "k" is used. The algorithm achieves this by identifying decision boundaries for each data point, and each data point is labeled based on where it fits inside those choice boundaries. Continuous data is required for this.

Using the K-Nearest Neighbor technique, the algorithm would be trained using a training set of continuous variables and linked to the outcome of being either churned or not. Once the algorithm has been trained on the dataset to determine which continuous factors define a customer as having churned, the model's performance will be evaluated on the test data set. As a result, a score for model accuracy will be generated. The objective is to create an algorithm that, without overfitting or underfitting the model, achieves model accuracy of at least 95%. One of the other expected results is the organization of test results based on their closest neighbors.

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

The K-Nearest Neighbor method's main premise is that comparable things will be found close together. This means that an unlabeled data point will reside close to a similar labeled data point, allowing it to be identified based on the similarity in attributes.

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

* The necessary programs and libraries for the k-NN classification are listed below, along with information on how they will help with 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. 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 classification method from part A1.**

For the categorization to function properly, the variables that are present (Yes or No) would be transformed to (0/1). Furthermore, the data will need to be verified for duplicate or missing data points, and if any are found, they will need to be properly handled (John Sullivan, 2018).

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

The categorical target variable will be "churn," as the objective is to determine the factors that impact whether a client will be lost (Yes or No). The initial set of categorical variables will include the following: Online Security, Online Backup, Techie, Contract, Tablet, Marital, Gender, Multiple, Device Protection, Internet Service, Phone, and Tech Support.

The following is a list of the initial continuous variables to be used: Children, age, income, email, bandwidth outages per week, contacts, yearly equipment failure, population, and monthly charge should all be taken into account.

The telecommunication company survey questions are categorized as discrete ordinal variables and may have an impact on a customer's decision to churn if necessary. The following identifiers will be assigned to the items, which have been officially classified in the dataset as Items 1 through 8, respectively. (Item1':'Timely response',' Item2':'Timely fixes',' Item3':'Timely replacements',' Item4':'Reliability',' Item5':'Options',' Item6':'Respectful response',' Item7':'Courteous exchange',' Item8':' Evidence of active listening)

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

Prior to completing the analysis, the data must be ready. Making sure that none of the columns have any missing data is the initial step. The next thing we should do is make sure there are no duplicates of any of the data in the columns. Additionally, we want to confirm that no columns or rows are duplicated, so we will check that and make sure the outcome is (False).

The dataset contains a number of variables that were determined to be useless for the logistic analysis, such as customer demographics that cannot be altered and are linked to the interaction and location of the consumer, so those columns should be removed.

Working with the data is now easier as a result. Any (yes or no) or other categorical alternatives need to be converted to numerical values in order to convert the categorical variables to numerical values. The survey columns also require renaming to provide a clearer comprehension and determination of applicable factors (Peter Grant, 2019).

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

The cleaned dataset is provided with the task submission named;



The below codes were executed in the creation of the cleaned and prepared dataset.

|  |
| --- |
| 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  from sklearn import datasets  from sklearn import preprocessing  from sklearn.neighbors import KNeighborsClassifier  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  pd.set\_option('display.max\_columns', None)  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()  # check if any rows are duplicated - looking for False  df.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()  #overview of descriptive statistics  df.describe()  #rename survey columns for easier identification  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)  #verify columns were renamed correctly  df.head()  #change yes/no to 1/0  df = df.replace(to\_replace = ['Yes','No'],value = [1,0])  #ensure values were changed  df.head()  # drop non-numeric variables that are less relevant  df = df.drop(['Marital', 'Gender','Contract','InternetService','PaymentMethod'], axis=1)  df.head()  print(list(df.columns))  #create histograms of both categorical and continuous variables  df[['Children', 'Age', 'Income', 'Churn', 'Outage\_sec\_perweek', 'Email', 'Contacts', 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year', 'Timely response', 'Timely fixes', 'Timely replacements', 'Reliability', 'Options', 'Respectful response', 'Courteous exchange', 'Evidence of active listening']].hist()  plt.savefig('Churn\_plot,jpg')  plt.show()  #create scatterplots to look for correlations  sns.scatterplot(x=df['MonthlyCharge'],y=df['Churn'],color='blue')  plt.show();  sns.scatterplot(x=df['Income'],y=df['Churn'],color='blue')  plt.show();  sns.scatterplot(x=df['Income'],y=df['MonthlyCharge'],color='blue')  plt.show();  #create scatterplots for numeric variables to view distributions and look for relationships  Churn\_numeric=df[['Children','Age','Income','Outage\_sec\_perweek','Contacts','Yearly\_equip\_failure','Tenure','MonthlyCharge','Bandwidth\_GB\_Year',  'Email','Respectful response','Courteous exchange','Evidence of active listening']]  pd.plotting.scatter\_matrix(Churn\_numeric,figsize=[15,15]);  #use a count plot to view featues of binary variables  plt.style.use('ggplot')  plt.figure()  sns.countplot(x='Phone',hue='Churn',data=df,palette='RdBu')  plt.show  plt.figure()  sns.countplot(x='Children',hue='Churn',data=df,palette='RdBu')  plt.show  plt.figure()  sns.countplot(x='Respectful response',hue='Churn',data=df,palette='RdBu')  plt.show  #move Churn target variable to end of columns  df=df[['Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Email', 'Contacts', 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year', 'Timely response', 'Timely fixes', 'Timely replacements', 'Reliability', 'Options', 'Respectful response', 'Courteous exchange', 'Evidence of active listening', 'Churn']]  df.head()  #export prepared dataset  df.to\_csv('prepared\_d209task1.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).**

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. To ensure that the outcomes could be repeated, a seed was planted.

The KNN model's n-nearest neighbor parameter was set to 7, instructing it to consider the seven closest values. These are taken into account when classifying. After fitting the data to the KNN model, the results were predicted using the data from earlier. The appendix contains the corresponding code.

The files for the train/test split have been submitted along with the data files.

Graphical user interface, text

Description automatically generated

**2. Describe the analysis technique you used to appropriately analyze the data. Include**

**screenshots of the intermediate calculations you performed.**

The initial KNN model had an accuracy score of 70.8%.

A picture containing table

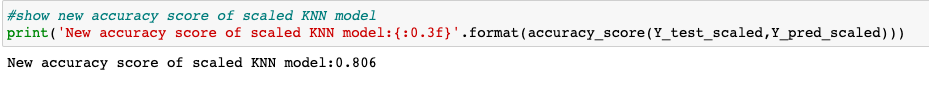
Description automatically generated

The metrics for the KNN model are seen below:

Table

Description automatically generated

The accuracy is then scaled to determine if it stays the same, improves, or degrades. The data is once more divided into train and test, and the KNN model is once more fitted. The result is a new model accuracy score of 80.6%.



Table

Description automatically generated

The scaled data is then used to create a confusion matrix that contrasts true and false negatives as well as true and false positives.

A picture containing table

Description automatically generated

Scaling the model decreased model performance. Accuracy moved from 0.7085 to 0.806 while

precision moved from 0.78 to 0.84.

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

|  |
| --- |
| #set predictor variables and target variable  x=df.drop('Churn',axis=1).values  y=df['Churn'].values  from sklearn.metrics import accuracy\_score  from sklearn.model\_selection import cross\_val\_score, train\_test\_split  #set seed in order to reproduce  SEED=1  #create training and test datasets  X\_train,X\_test,Y\_train,Y\_test = train\_test\_split(x,y,test\_size=0.20,random\_state=SEED)  #export test and training data  X\_train.tofile(r'C:\Users\Ibrahim\Desktop\churn\_Xtrain.csv',sep=',')  X\_test.tofile(r'C:\Users\Ibrahim\Desktop\churn\_Xtest.csv',sep=',')  Y\_train.tofile(r'C:\Users\Ibrahim\Desktop\churn\_Ytrain.csv',sep=',')  Y\_test.tofile(r'C:\Users\Ibrahim\Desktop\churn\_Ytest.csv',sep=',')  #begin KNN model  knn=KNeighborsClassifier(n\_neighbors=7)  #fit data to KNN model  knn.fit(X\_train,Y\_train)  #predict outcomes from test data  Y\_pred=knn.predict(X\_test)  #show initial accuracy score of KNN model  print('Initial accuracy score of KNN model: ',accuracy\_score(Y\_test,Y\_pred))  #compute classification metrics  print(classification\_report(Y\_test,Y\_pred))  #scale data  from sklearn.preprocessing import StandardScaler  from sklearn.pipeline import Pipeline  from sklearn.metrics import accuracy\_score  steps=[('scaler',StandardScaler()),('KNN',KNeighborsClassifier())]  pipeline=Pipeline(steps)  #split data  X\_train\_scaled,X\_test\_scaled,Y\_train\_scaled,Y\_test\_scaled=train\_test\_split(x,y,test\_size=0.20,random\_state=SEED)  #scale data with pipeline  KNN\_scaled=pipeline.fit(X\_train\_scaled,Y\_train\_scaled)  #predict from scaled data  Y\_pred\_scaled=pipeline.predict(X\_test\_scaled)  #show new accuracy score of scaled KNN model  print('New accuracy score of scaled KNN model:{:0.3f}'.format(accuracy\_score(Y\_test\_scaled,Y\_pred\_scaled)))  #compute new classification metrics after scaling  print(classification\_report(Y\_test\_scaled,Y\_pred\_scaled))  #import confustin matrix from sklearn  from sklearn.metrics import confusion\_matrix  cf\_matrix=confusion\_matrix(Y\_test,Y\_pred)  print(cf\_matrix)  #visualize confustion matrix  group\_names=['True Neg','False Pos','False Neg','True Pos']  group\_counts=["{0:0.0f}".format(value) for value in cf\_matrix.flatten()]  group\_percentages=["{0:.2%}".format(value) for value in cf\_matrix.flatten()/np.sum(cf\_matrix)]  labels=[f"{v1}\n{v2}\n{v3}" for v1,v2,v3 in  zip(group\_names,group\_counts,group\_percentages)]  labels=np.asarray(labels).reshape(2,2)  sns.heatmap(cf\_matrix,annot=labels,fmt='',cmap='Blues') |
|  |

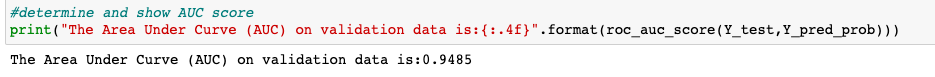
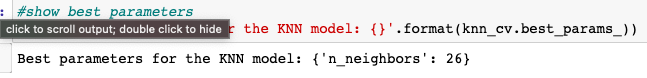
**Part V: Data Summary and Implications**

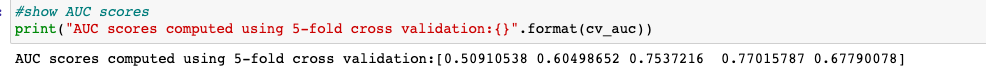
**E. Summarize your data analysis by doing the following:**

**1. Explain the accuracy and the area under the curve (AUC) of your classification model.**

We used cross-validation to analyze the training data and determine the optimal KNN model parameters. The main choices in our circumstance were "n-neighbors: 26." The KNN model's top rating was 0.742. The validation data were used to calculate the Area Under Curve (AUC), which yielded a score of 0.9485.

The scores that were obtained using the 5-fold cross-validation approach are as described in the following: [0.50910538 0.60498652 0.7537216 0.77015787 0.67790078]. In 94.85% of occurrences, the KNN model generates a true positive result.

Text

Description automatically generated

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

The model's accuracy score is 0.806, its precision score is 0.84, its recall score is 0.9, its F1 score is 0.87, and its AUC score is 0.9485. The algorithm demonstrates that it can forecast a customer's status properly 90% of the time and is able to predict that a client won't leave 84% of the time.

To put it another way, this model is effective at classifying data and can generate findings that are genuinely positive with a high degree of confidence. As a result, employing the discovered characteristics in the future will be a useful first step toward developing exact predictions about whether a client will churn.

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

Finding K for the K-Nearest Neighbor technique is the analysis's main shortcoming. The analyst selects K at random in order to more accurately categorize the data. Although the analyst might think that a higher K value will provide a model that is more accurate, in reality, it might cause the model to overfit as a result of under-fitting, the user could also select a K value that is too low.

All of them result in a model that will not classify data precisely or accurately. K = 26 served as the foundation in our example. K = <25 was actually a superior fit, as we discovered through model scaling and modification. As a result, running the model would take less time and use less memory.

**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.**

The researcher advises the telecommunications business to further assess the variables to ascertain which are the most significant predictor variables for a client being lost. Even if the accuracy and precision score of the current model is good (0.80.6), more analysis might be done to establish whether the model was overfitting or underfitting the data.

Performing the study once more with K = 3 can result in a model that is more precise and takes less time to categorize consumers and predict attrition. The staff of the telecommunications firm can concentrate on those variables once the relevant predictor variables have been identified with high accuracy and precision in order to lessen the possibility of a customer leaving.

**You can view the session using the following link:**

Session 1 Video  
<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=eb481e65-25ad-466a-8536-af1100d1f421>

Session 2 Video   
<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7ab76c98-7e7b-4372-807e-af1100e04d99>

**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.