D209 Data Mining I: Performance Assessment

Task 1: Classification Analysis

Jill Rivera

Western Governors University

Table of Contents

[I. Overview 3](#_Toc106346599)

[II. Method Justification 3](#_Toc106346600)

[III. Data Preparation 4](#_Toc106346601)

[IV. Analysis 12](#_Toc106346602)

[V. Data Summary and Implications 23](#_Toc106346603)

[VI. Scholarly References 25](#_Toc106346604)

[VII. Third-Party Code References 25](#_Toc106346605)

[VIII. Table of Figures 26](#_Toc106346606)

# **Overview**

1. Customer churn, or when a customer discontinues services with a company, is a problem that every business in the service industry deals with. This analysis will dive deeper into a data set for a telecommunications company, with the objective of understanding which customers are at a higher risk of churning from their company.
   1. **Research Question**: The data set and associated data dictionary contain 10,000 customer records with 50 attributes used to describe them. The question of whether a new customer can be classified using certain data points in the data set will be analyzed. The method of analysis chosen is the K-Nearest Neighbors (KNN) machine learning algorithm.
   2. **Objectives**: This analysis will classify a new control customer as either at risk or not of churning based on given metrics. For this, the features of Monthly Charge and Tenure will be used to build a model to classify our new customer and test the model’s accuracy.

# **Method Justification**

1. Classification analysis classifies a new data point based on its similarity to neighboring data points of attributes. Using the data points of other features in a data set, the new value can be categorized as a part of group of similar values.
   1. **Method (KNN)**: K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that is perfect for this analysis. KNN is a widely used classification method used to calculate the distance between your target point and similar data points in the model. This method uses a straight-line or Euclidean distance to solve the problem. (Harrison, 2018)
   2. **Assumption**: One assumption of the KNN model is that if data points are close together, then they are highly similar. This method uses the distance between data points to determine this. Conversely, if the data points are far from each other, then they are not similar.
   3. **Tools**: This analysis will be conducted using the Python programming language in a Jupyter Notebook. Both are open-source platforms that are easy to use and powerful in solving these simple machine learning algorithms. The following table displays the Python libraries that will be used in this analysis and their purpose.

|  |  |
| --- | --- |
| **Package/ Library** | **Usage** |
| Pandas | Data frame reading/writing |
| Numpy | Data frame manipulation |
| Matplotlib | Graphing variable independence |
| Sklearn.feature\_selection: SelectKBest.f\_classif | Predictive modeling package |
| Sklearn.preprocessing: Scale | Confusion matrix creation |
| Sklearn.model\_selection: Train\_test\_split | Performance curve creation |
| Sklearn.pipeline: Pipeline | Building classification trees |

Figure : Python Packages and Libraries

# **Data Preparation**

1. Perform data preparation for the chosen data set by doing the following: The data set will be imported into a Jupyter Notebook for analysis. The data will be cleaned, explored, and wrangled before the classification model is created.
   1. **Objective**: KNN analysis will be used to determine the customer records that are closest in similarity to the new customer created as a control. To do this, the data needs to be complete with no missing or duplicated data. The categorical variable of Churn will be changed from an object type with yes/no values to a Boolean type with true/false values. This will treat the variable as numeric for the model analysis.
   2. **Variable Statistics**: The target variable for this analysis is the churn feature. This variable is categorical and has yes/no values. Of the 10,000 customer records, 7,350 have not churned and 2,650 have left the company. This is visualized in a bar graph.

Chart, bar chart

Description automatically generated

Figure : Churn Variable Graph

The explanatory features that will be used are tenure and monthly charge. Both explanatory variables are continuous numerical data types. The range and mean of each are shown in the table.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Range** | **Mean** |
| Tenure | 1.0025934 – 71.99928 | 34.53 +/- 26.44 |
| Monthly Charge | 79.97886 – 290.160419 | 172.62 +/- 42.94 |

Figure : Explanatory Variable Statistics

When comparing the target variable to the explanatory variables, the statistics can be seen. The average customer who left the company was paying $199.30 monthly and had only 13 months of service. While those who have stayed only pay $163.01 on average and have been with the company for 42 months. This shows that the customers who are paying more are leaving early on into their service.

**Graphical user interface, application

Description automatically generated with medium confidence**

Figure : Numerical Mean Data vs Churn

* 1. **Data Cleaning**: The Python packages and libraries used for the analysis are imported and configured.

Text

Description automatically generated

Figure : Importing Standard Libraries

Graphical user interface, text, application

Description automatically generated

Figure : Importing and Configuring SciKit Learn (Pedregosa, Varoquaux, & Gramfort, 3.3. Metrics and scoring: quantifying the quality of predictions, 2022)

The CSV file is read using the Pandas read\_csv method.

Text

Description automatically generated with low confidence

Figure : Loading the Data Set

The features of the data set are listed, and the shape of the data frame is displayed. There are a total of 50 features and 10,000 customer records in the data set.

Text, letter

Description automatically generated

Figure : Initial Data Frame Structure

A new data frame using only the variables that are used in the analysis is created, and the first five rows are displayed.

Graphical user interface, text, application

Description automatically generated

Figure : New Data Frame Creation

The data frame is checked for null values using the isnull and sum methods. We find no missing values in the data.

Table

Description automatically generated with medium confidence

Figure : Null Value Detection

The rows are checked to make sure that there are no duplicated rows of data. Using the duplicated and any methods, we find no rows are copies of each other.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure : Duplicate Detection

Outlying values are checked using the z-score of each variable. This proved to find no outlying values in the data.

Graphical user interface, text, application

Description automatically generated

Figure : Outlying Value Detection

The data is scaled to ensure there is an even number of true and false values.

A picture containing graphical user interface

Description automatically generated

Figure : Scaling Features

The clean data set consists of 10,000 customer records and the churn, tenure, monthly charge, z monthly charge, and z tenure variables for classification analysis. A visualization of the features, their data type and metrics is shown.

Graphical user interface, text

Description automatically generated

Figure : Clean Data Set

* 1. **Data Wrangling**: The categorical feature of churn has yes/no values initially. These values will be changed to Boolean values of true and false. Boolean values are considered numerical or binary for such calculations. Using the replace and astype methods, the values are changed as shown.

Text

Description automatically generated

Figure : Data Wrangling

* 1. **Extract**: A copy of the clean data set is exported to a CSV file labeled 209clean\_task1.csv. This file is attached as a submission for this assessment.

**Graphical user interface, text

Description automatically generated**

Figure : Extract Clean Data Set

# **Analysis**

1. The classification analysis is performed by first defining the target and primary features and then splitting the data into training and testing groups. The control customer for the analysis is created to test the model. The model is built and run with the training data only and then again using all the data. The model is analyzed through a confusion matrix and the area under the curve is evaluated through the receiver operating curve graph.
   1. **Train/Test Split**: The data frame will be split into training and testing data groups. This will allow the model to be built on the training data and the accuracy can be tested on the testing data. First, the primary features and target data are defined in variables X and y.

**Text

Description automatically generated**

Figure : Defining Data Labels

The data is split into these groups using Scikit-learn’s train\_test\_split function, where 70% of the data is for training and the other 30% is reserved for testing.

Text

Description automatically generated

Figure : Train/Test Split of Data (Pedregosa, Varoquaux, & Gramfort, sklearn.model\_selection.train\_test\_split, 2022)

A copy of each data set is exported to a CSV file for submission. The labels of these files include 209trainData\_task1.csv and 209testData\_task1.csv.

Graphical user interface, text

Description automatically generated

Figure : Exporting Train/Test Data

* 1. **Classification Analysis**: The KNN analysis will determine if a new customer we create as a control will classify them as churn true or false, meaning more or less likely to leave the provider. First, the new customer is defined. The new customer will have a tenure value of 1.0 and a monthly charge value of 175.00.

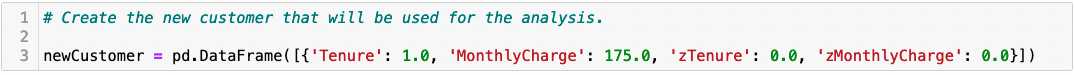


Figure : New Customer Defined

A scatter plot of monthly charge compared to tenure is displayed using the training data and Seaborn.

Chart, scatter chart

Description automatically generated

Figure : Training Data Scatter Plot (Waskom, 2022)

A scatter plot of the primary features is created using the testing data set.

Chart, scatter chart

Description automatically generated

Figure : Testing Data Scatter Plot (Waskom, 2022)

The KNN model is created using the KNeighborsClassifier function from Scikit-learn.

Graphical user interface, text, application

Description automatically generated

Figure : KNN Model Creation (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022)

Predicted and observed data is defined.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure : Predicted and Observed Data Defined

Incorrectly predicted data is defined and counted. There are 523 incorrectly predicted data points in the model.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure : Incorrectly Predicted Data

The KNN score is calculated for the model using the score method. This is the accuracy score for the model created.

Graphical user interface, text

Description automatically generated with medium confidence

Figure : KNN Score

Hyperparameter tuning is performed on the model to find the number of nearest neighbors that should be used in the model. Running this shows that when k is equal to seven, the accuracy of 82% is achieved.

Text

Description automatically generated

Figure : Hyperparameter Tuning (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022)

The new customer data is scaled between the primary features to distribute the values equally.

A picture containing text

Description automatically generated

Figure : Scaling Customer Data

The Scikit-learn NearestNeighbors function is used to compute the KNN analysis on the training data. Setting the k to seven as determined in the hyperparameter tuning, yields the seven customer records that are closest to the new customer. These all indicate that the new customer should be classified as churn false.

Graphical user interface, text, application

Description automatically generated

Figure : KNN Model Code Using Training Data (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022)

Table

Description automatically generated

Figure : KNN Training Model Output

The model is re-trained using all the data and k as seven again. This displays the seven customer records which are closest to the new customer’s metrics.

Table

Description automatically generated

Figure : Final KNN Model (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022)

This data is plotted in a scatter plot, with the new customer indicated as a star.

Chart, scatter chart

Description automatically generated

* 1. **Analysis Technique**: A confusion matrix is created to assess the sensitivity (portion of the positive class that was correctly classified) and specificity (portion of the negative class that was correctly classified) of the model. This matrix shows that of 3000 predictions, 2477 were correct making the accuracy of the model 82.6%.

Chart, treemap chart

Description automatically generated

Figure : Confusion Matrix

Metrics for the confusion matrix verify this accuracy rate. This also verifies that there is a 17% chance of error. When put in terms of the new customer created to test the model, there is an 83% chance of them having the value of churn = True and a 17% chance of them having the value of churn = False.

Text

Description automatically generated with medium confidence

Figure : Confusion Matrix Metrics

Text

Description automatically generated

Figure : Confusion Matrix Code

A Receiver Operating Curve (ROC) was used to evaluate the model’s accuracy in predicting outcomes. This technique plots the true positive rate (TPR) against the false positive rate (FPR) to separate the ‘signal’ from the ‘noise’. (Bhandari, 2020) The Area Under the Curve (AUC) is a calculation used to measure the ability of the model to distinguish between the positive and negative classes. The AUC value ranges between 0 to 1, with the value of 1 meaning that there is a high chance of positive values being distinguished from negative values. When analyzing the graph of the ROC, the higher X-axis value present shows a higher number of false positives than true negatives in the model. On the other hand, a higher Y-axis value shows a larger number of true positives than false negatives in the model.

Chart, line chart

Description automatically generated

Figure : Receiver Operating Curve

Graphical user interface, text, application

Description automatically generated

Figure : ROC Code (Pedregosa, Varoquaux, & Gramfort, Receiver Operating Characteristic (ROC), 2022)

# **Data Summary and Implications**

1. **Findings and Assumptions**:

The final model is run using both the training and testing data sets. The accuracy, results and limitations of the model are outlined. Recommendations on the practical use of this model are given.

* 1. **Accuracy**: The final model proved to maintain an accuracy of 82% using k as seven to determine the seven closest customers with similar data points to the control customer. This was verified using the AUC in the ROC graph and by calculating the KNN score.
  2. **Results**: The final seven similar customer records to the control customer are displayed. Five out of seven of the records indicate a true churn value. This means that the control customer is at a high risk (71% chance) of leaving the provider. This could be due to the monthly payment being higher than the average calculated for the whole data set. The control data can be manipulated, and the analysis run another time using lower parameters to confirm this theory.

A picture containing text, receipt

Description automatically generated

Figure : Final Neighbors Results

* 1. **Limitations**: A limitation to this analysis is the variable selection. In choosing tenure, a new customer will always have a low value for this attribute. For the KNN computation however, it is not advisable to test a parameter with such a low value. A better variable selection could be made to compare the customer’s bandwidth usage or amount of outage time they experience instead to produce a better model for new customers. This model, however, will prove useful in classifying existing customers as likely or unlikely to churn.
  2. **Recommendations**: Additional analysis is recommended using this model with other variables such as bandwidth or outage time experienced to see if new customers can be evaluated more accurately. Using this model to evaluate existing customers is advised to determine if their monthly charge is too high based on their tenure and propensity to churn. An existing customer flagged by the model could be offered a discount on their monthly service to decrease their risk of churning and increase their tenure.

# **Scholarly References**

Bhandari, A. (2020, June 16). *AUC-ROC Curve in Machine Learning Clearly Explained*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Harrison, O. (2018, September 10). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Retrieved from Towards Data Science: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

# **Third-Party Code References**

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2022, June 14). *3.3. Metrics and scoring: quantifying the quality of predictions*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/model\_evaluation.html

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2022, June 14). *Receiver Operating Characteristic (ROC)*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_roc.html

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2022, June 14). *sklearn.model\_selection.train\_test\_split*. Retrieved from Scikiit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.train\_test\_split.html

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2022, June 14). *sklearn.neighbors.KNeighborsClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Waskom, M. (2022, June 14). *seaborn.scatterplot*. Retrieved from Seaborn: https://seaborn.pydata.org/generated/seaborn.scatterplot.html

# **Table of Figures**

[Figure 1: Python Packages and Libraries 4](#_Toc106346562)

[Figure 2: Churn Variable Graph 5](#_Toc106346563)

[Figure 3: Explanatory Variable Statistics 6](#_Toc106346564)

[Figure 4: Numerical Mean Data vs Churn 6](#_Toc106346565)

[Figure 5: Importing Standard Libraries 7](#_Toc106346566)

[Figure 6: Importing and Configuring SciKit Learn (Pedregosa, Varoquaux, & Gramfort, 3.3. Metrics and scoring: quantifying the quality of predictions, 2022) 7](#_Toc106346567)

[Figure 7: Loading the Data Set 8](#_Toc106346568)

[Figure 8: Initial Data Frame Structure 8](#_Toc106346569)

[Figure 9: New Data Frame Creation 9](#_Toc106346570)

[Figure 10: Null Value Detection 9](#_Toc106346571)

[Figure 11: Duplicate Detection 9](#_Toc106346572)

[Figure 12: Outlying Value Detection 10](#_Toc106346573)

[Figure 13: Scaling Features 10](#_Toc106346574)

[Figure 14: Clean Data Set 11](#_Toc106346575)

[Figure 15: Data Wrangling 11](#_Toc106346576)

[Figure 16: Extract Clean Data Set 12](#_Toc106346577)

[Figure 17: Defining Data Labels 12](#_Toc106346578)

[Figure 18: Train/Test Split of Data (Pedregosa, Varoquaux, & Gramfort, sklearn.model\_selection.train\_test\_split, 2022) 13](#_Toc106346579)

[Figure 19: Exporting Train/Test Data 13](#_Toc106346580)

[Figure 20: New Customer Defined 14](#_Toc106346581)

[Figure 21: Training Data Scatter Plot (Waskom, 2022) 14](#_Toc106346582)

[Figure 22: Testing Data Scatter Plot (Waskom, 2022) 15](#_Toc106346583)

[Figure 23: KNN Model Creation (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022) 15](#_Toc106346584)

[Figure 24: Predicted and Observed Data Defined 15](#_Toc106346585)

[Figure 25: Incorrectly Predicted Data 16](#_Toc106346586)

[Figure 26: KNN Score 16](#_Toc106346587)

[Figure 27: Hyperparameter Tuning (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022) 17](#_Toc106346588)

[Figure 28: Scaling Customer Data 17](#_Toc106346589)

[Figure 29: KNN Model Code Using Training Data (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022) 18](#_Toc106346590)

[Figure 30: KNN Training Model Output 18](#_Toc106346591)

[Figure 31: Final KNN Model (Pedregosa, Varoquaux, & Gramfort, sklearn.neighbors.KNeighborsClassifier, 2022) 19](#_Toc106346592)

[Figure 32: Confusion Matrix 20](#_Toc106346593)

[Figure 33: Confusion Matrix Metrics 21](#_Toc106346594)

[Figure 34: Confusion Matrix Code 22](#_Toc106346595)

[Figure 35: Receiver Operating Curve 23](#_Toc106346596)

[Figure 36: ROC Code (Pedregosa, Varoquaux, & Gramfort, Receiver Operating Characteristic (ROC), 2022) 23](#_Toc106346597)

[Figure 37: Final Neighbors Results 24](#_Toc106346598)