**D209 Essay Part 1**

**CLASSIFICATION ANALYSIS**

Darien Nguyen

Western Governors University

D209: Classification Analysis

January 21, 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:

• k-nearest neighbor (KNN)

• Naive Bayes

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.

What factors are most likely to predict customer churn in a telecommunications company and can a Naive Bayes classification model be built to accurately predict customer churn using those factors?

This question is relevant to a real-world organizational situation, as it addresses the problem of customer churn in the telecommunications industry, which can have significant financial consequences for companies. Additionally, the question proposes using the Naive Bayes classification method to analyze the data, which is appropriate method for predicting categorical outcomes. The goal of the data analysis could be to identify the key factors that contribute to customer churn and to build a Naive Bayes classification model that can accurately predict customer churn based on those factors.

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.

2. Summarize one assumption of the chosen classification method.

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 Gaussian Naive Bayes algorithm is a probabilistic classifier that applies Bayes' theorem to compute the probability of a particular class given a set of features. The algorithm assumes that the features are independent from each other, and that the data follows a Gaussian distribution.

One assumption of the Naive Bayes algorithm is that the features are independent from each other.

The packages or libraries used in the provided code are pandas, sklearn. naive\_bayes and sklearn.model\_selection. The pandas library is used to read in and manipulate the data set, the sklearn.naive\_bayes library is used to import the Gaussian Naive Bayes algorithm and the sklearn.model\_selection library is used to split the data into training and test sets.

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.

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.

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

4. Provide a copy of the cleaned data set.

We have performed several steps of data preparation for the chosen data set. One of the goals of our data preprocessing was to handle any missing values, which could affect the performance of the classification method outlined in part A1. To accomplish this, we dropped any columns that had missing values and dropped any rows with missing values in the remaining columns. Additionally, we converted the "Yes" and "No" values to 1s and 0s and one-hot encoded the categorical variables. Another goal of our data preprocessing was to handle any outliers that could impact the performance of the classification model. To achieve this, we calculated the Z-scores of all the columns in the data set, set a threshold for the Z-scores, identified the rows with Z-scores above the threshold, and dropped those rows. Finally, we split the preprocessed data into training and test sets.

To classify each variable, we can inspect the data type and the values of each column to determine if they are continuous or categorical. The categorical variables can be identified by the presence of non-numeric data, such as "Male" and "Female" in the case of Gender, or by the presence of numerical values that represent categories, such as 0 and 1 in the case of Contract\_Month-to-month. The continuous variables can be identified by their numerical data type and the presence of numerical values that can take any value within a certain range, such as Income and Age.

We have identified that the following variables are categorical:

* Children
* Techie
* Tablet
* Phone
* Multiple
* Item1
* Item2
* Item3
* Item4
* Item5
* Item6
* Item7
* Item8
* churn
* Gender\_Male
* Gender\_Nonbinary
* Contract\_Month-to-month
* Contract\_One year
* Contract\_Two Year
* InternetService\_DSL
* InternetService\_Fiber Optic
* InternetService\_None

And the following variables are continuous:

* Age
* Income
* Outage\_sec\_perweek
* Yearly\_equip\_failure

The following steps were taken to prepare the data for analysis in this code:

1. Read in the data set: df = pd.read\_csv('churn\_clean.csv')
2. Drop columns with missing values: df = df.drop(columns=['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Email', 'Contacts', 'City', 'State', 'Marital', 'PaymentMethod', 'PaperlessBilling'])
3. Drop any remaining missing values: df = df.dropna()
4. Convert target column "Churn" to binary values: df["churn"] = df["Churn"].apply(lambda x: 1 if x == "Yes" else 0), df.drop("Churn", axis = 1, inplace = True)
5. One-hot encode categorical variables: df = pd.get\_dummies(df, columns=['Gender'], prefix='Gender', drop\_first=True), df = pd.get\_dummies(df, columns=['Contract', 'InternetService'])
6. Replace categorical values with binary values: df.replace(to\_replace={'Yes':1, 'No':0}, inplace=True)
7. Check for missing values: missing\_values = df.isnull().sum(), print(missing\_values)
8. Compute the Z-scores: z\_scores = df.apply(zscore)
9. Identify and drop rows with Z-scores above a threshold: threshold = 3, outliers = np.where(np.abs(z\_scores) > threshold), outlier\_indices = list(set(outliers[0])), df = df.drop(df.index[outlier\_indices])
10. Split the data into training and test sets: X = df.drop(['churn'], axis=1), y = df['churn'], X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

With the data properly prepared, the Gaussian Naive Bayes model was then fit to the training data, used to make predictions on the test data, and evaluated for accuracy using accuracy\_score from the sklearn.metrics library.

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).

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

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

We start by importing the necessary libraries including pandas, numpy, scikit-learn's Gaussian Naive Bayes and train\_test\_split modules, and scipy's zscore. Then, we read the data set into a pandas dataframe, df.

Next, we drop any columns in the dataframe that contain missing values and also remove any rows that have missing values. We then convert the "Churn" column into a binary target column named "churn". We then use pandas' get\_dummies method to one-hot encode the categorical variables in the dataframe.

We then replace the "Yes" and "No" values with 1s and 0s respectively and calculate the Z-scores of all the columns in the dataframe using the apply method and zscore from scipy. We then set a threshold for the Z-scores, identify any rows with Z-scores above this threshold, and remove these outliers.

Finally, we split the cleaned data into training and test sets, X\_train and X\_test, and y\_train and y\_test respectively, using the train\_test\_split function from scikit-learn. The training data is used to fit a Gaussian Naive Bayes model, nb, which is then used to make predictions on the test data. The accuracy of the model is then evaluated using the accuracy\_score function from scikit-learn and is printed.

Chart, bar chart

Description automatically generated

# Compute the Z-scores of all columns in the dataframe

z\_scores = df.apply(zscore)

# Set a threshold for the Z-scores

threshold = 3

print("First 10 rows of Z-scores DataFrame:")

print(z\_scores.head(10))

# Identify rows with Z-scores above the threshold

outliers = np.where(np.abs(z\_scores) > threshold)

# Get the indices of the rows with outliers

outlier\_indices = list(set(outliers[0]))

print("Indices of rows with outliers:")

print(outlier\_indices)

# Drop the rows with outliers

df = df.drop(df.index[outlier\_indices])

print("First 10 rows of the updated dataframe:")

print(df.head(10))

# Split the data into training and test sets

X = df.drop(['churn'], axis=1)

y = df['churn']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create the Naive Bayes model

nb = GaussianNB()

nb.fit(X\_train, y\_train)

# Predict using the test set

y\_pred = nb.predict(X\_test)

# Evaluate the model's accuracy

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy: ", accuracy)

# Evaluate the model's accuracy and AUC

accuracy = accuracy\_score(y\_test, y\_pred)

auc = roc\_auc\_score(y\_test, y\_pred)

print("Accuracy: ", accuracy)

print("AUC:", auc)

# Plot the accuracy and AUC

width = 0.35

fig, ax = plt.subplots()

rects1 = ax.bar(1, accuracy, width, color='red')

rects2 = ax.bar(2, auc, width, color='blue')

ax.set\_ylabel('Score')

ax.set\_title('Accuracy and AUC Comparison')

ax.set\_xticks([1, 2])

ax.set\_xticklabels(['Accuracy', 'AUC'])

plt.show()

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.

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

3. Discuss one limitation of your data analysis.

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 accuracy of the classification model is 0.7799642218246869, which means that the model correctly predicts the class of 77.99% of the samples in the test set. The area under the curve (AUC) of the model is 0.817020963603252, which represents the model's ability to distinguish between positive and negative classes. The AUC is a metric used to evaluate the performance of binary classification models, and a value of 1.0 represents a perfect classifier, while a value of 0.5 represents a classifier that performs no better than random.

The results of the classification analysis show that the model has a relatively high accuracy and AUC, which suggests that it is able to perform well in classifying the samples in the test set.

One limitation of the data analysis is that it is based on a single evaluation metric, accuracy, and a single performance metric, AUC. While these metrics provide valuable information about the performance of the model, they do not provide a complete picture of the model's performance. For example, it is possible that the model has a high accuracy but a low precision, meaning that it correctly identifies many of the positive samples but also falsely identifies many negative samples as positive.

Based on the results and implications of the data analysis, it can be recommended that the telecommunication company take the following course of action:

1. Utilize the classification model to identify customers who are at high risk of churning. By focusing on these customers, the company can take proactive measures to retain them and prevent them from leaving.
2. Develop targeted retention strategies based on the insights gained from the data analysis. For example, if the analysis revealed that customers who have been with the company for a long time are more likely to churn, the company could offer incentives for customer loyalty.
3. Continuously monitor and update the classification model to ensure it remains relevant and accurate. As customer behavior and preferences change, the model may need to be adjusted to reflect these changes.
4. Consider collecting additional data and incorporating it into the model. This can help improve its accuracy and provide a more comprehensive understanding of the factors that contribute to customer churn.
5. Encourage customer feedback and engage in regular communication with customers. By doing so, the company can gain a deeper understanding of customer needs and preferences, which can help inform retention strategies and improve customer satisfaction.

Overall, the telecommunication company should take a proactive and data-driven approach to reducing customer churn and improving customer satisfaction. By utilizing the results of the data analysis and taking appropriate action, the company can retain valuable customers and improve its bottom line.

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.

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a00199d7-51f9-4b3b-90ef-afaa0042243d>

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.

No sources used.

H. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

No sources used.