Task 1

WGU | D209

D209

Narcisse, Laurie

2023

Contents

[Part I: Research Question 2](#_Toc155173717)

[A. Describe the purpose of this data mining report by doing the following: 2](#_Toc155173718)

[1. Propose one question relevant to a real-world organizational situation that you will answer using one of the following classification methods: 2](#_Toc155173719)

[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. 2](#_Toc155173720)

[Part II: Method Justification 3](#_Toc155173721)

[B. Explain the reasons for your chosen classification method from part A1 by doing the following 3](#_Toc155173722)

[1. Explain how the classification method you chose analyzes the selected data set. Include expected outcomes. 3](#_Toc155173723)

[2. Summarize one assumption of the chosen classification method. 3](#_Toc155173724)

[3. List the packages or libraries you have chosen for Python or R, and justify how each item on the list supports the analysis. 3](#_Toc155173725)

[Part III: Data Preparation 6](#_Toc155173726)

[C. Perform data preparation for the chosen data set by doing the following: 7](#_Toc155173727)

[1. Describe one data preprocessing goal relevant to the classification method from part A1. 7](#_Toc155173728)

[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 numeric or categorical. 7](#_Toc155173729)

[3. Explain each of the steps used to prepare the data for the analysis. Identify the code segment for each step. 7](#_Toc155173730)

[4. Provide a copy of the cleaned data set. 13](#_Toc155173731)

[Part IV: Analysis 13](#_Toc155173732)

[D. Perform the data analysis and report on the results by doing the following: 13](#_Toc155173733)

[1. Split the data into training and test data sets and provide the file(s). 13](#_Toc155173734)

[2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed. 13](#_Toc155173735)

[3. Provide the code used to perform the classification analysis from part D2. 19](#_Toc155173736)

[Part V: Data Summary and Implications 24](#_Toc155173737)

[E. Summarize your data analysis by doing the following: 24](#_Toc155173738)

[1. Explain the accuracy and the area under the curve (AUC) of your classification model. 24](#_Toc155173739)

[2. Discuss the results and implications of your classification analysis. 24](#_Toc155173740)

[3. Discuss one limitation of your data analysis. 24](#_Toc155173741)

[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. 25](#_Toc155173742)

[Part VI: Demonstration 25](#_Toc155173743)

[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. 25](#_Toc155173744)

[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. 25](#_Toc155173745)

# 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:

Question: Can customer churn be accurately predicted by examining the intricate interplay between demographic information and usage patterns, utilizing a classification method like k-Nearest Neighbor (KNN)?

### 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.

One primary goal of my data analysis is to unearth key demographic and usage patterns that play a substantial role in predicting customer churn. I aim to delve into the available dataset, focusing on specific socio-economic factors and behavioral metrics such as age, gender, income, and usage patterns. The objective is to pinpoint indicators within the data that strongly suggest whether a customer is likely to churn or remain with the service.

By identifying these influential factors, my analysis seeks to offer actionable insights that businesses can use to proactively address and mitigate customer churn. The emphasis is on understanding the intricate relationship between demographic information and usage patterns, empowering companies to tailor retention strategies, improve customer satisfaction, and potentially enhance overall business performance.

The feasibility of this goal is supported by the dataset at my disposal, which includes relevant variables like 'Churn,' 'Age,' 'Gender,' 'Income,' and various usage pattern metrics. Utilizing these variables alongside the KNN classification method, I aim to explore patterns and relationships that significantly contribute to predicting customer churn. This approach provides practical and actionable information for strategic decision-making.

I will consider the analysis successful if it identifies specific features demonstrating a statistically significant impact on customer churn prediction. This outcome will serve as a solid foundation for constructing an effective predictive model and guiding targeted business interventions.

It's important to acknowledge that while the analysis can reveal correlations, establishing causation would require additional experimental designs. Nevertheless, my goal is reasonable within the scope of leveraging the available data to inform strategies aimed at reducing customer 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.

The classification method I chose, k-Nearest Neighbors (KNN), analyzes the selected dataset by grouping similar data points based on their features and classifying new instances according to the majority class within their k-nearest neighbors. In the context of predicting customer churn, KNN evaluates the demographic information (age, gender) and usage patterns (frequency and duration of product or service usage, satisfaction ratings) to identify patterns associated with customers likely to churn. I expect the KNN model to successfully classify instances into churn or non-churn categories, providing insights into the factors influencing customer attrition.

I anticipate that the KNN model will capture non-linear relationships between features and churn, contributing to a nuanced understanding of customer behavior. The model should identify distinct clusters of customers with similar characteristics and predict their likelihood of churning. Additionally, the analysis should reveal the importance of specific features in predicting churn, offering actionable insights for targeted retention strategies. Overall, I expect a well-performing KNN model that enhances our organization's ability to proactively address customer churn based on a comprehensive understanding of demographic and usage-related patterns.

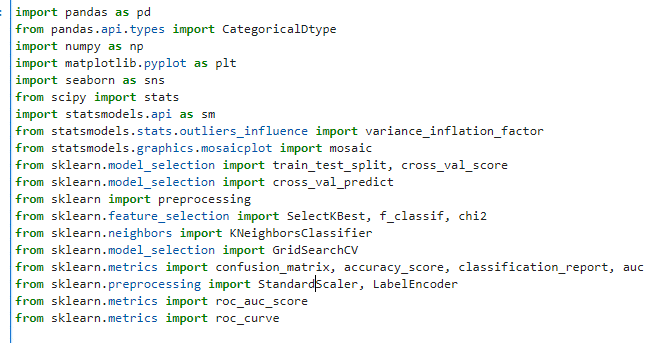
### 2. Summarize one assumption of the chosen classification method.

In my exploration of the k-nearest neighbor (KNN) method, a fundamental assumption underlying its functioning is that instances with similarities in the feature space tend to have similar labels. This foundational concept is pivotal to the model's predictive capacity, as it relies on the notion that neighboring data points share common characteristics and, therefore, are likely to belong to the same class. An essential consideration in implementing KNN is the selection of the parameter 'k,' representing the number of nearest neighbors considered during classification. The choice of 'k' is paramount as it directly influences the model's performance. Too few neighbors may result in sensitivity to outliers, while too many may lead to over smoothing and a lack of discrimination. Striking the right balance for 'k' is a critical aspect of fine-tuning the KNN model to achieve optimal predictive accuracy.

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

I will be using Python and the following libraries:

* pandas (import pandas as pd):
  + Pandas is a powerful data manipulation and analysis library for Python. It provides data structures like DataFrame and Series, making it easy to handle and analyze structured data.
* CategoricalDtype (from pandas.api.types):
  + CategoricalDtype is a specific data type within pandas that represents categorical data. It's useful for handling variables with a limited, fixed set of categories.
* numpy (import numpy as np):
  + NumPy is a numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.
* matplotlib.pyplot (import matplotlib.pyplot as plt):
  + Matplotlib is a plotting library for Python. The pyplot module provides a convenient interface for creating various types of plots and visualizations.
* seaborn (import seaborn as sns):
  + Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics.
* scipy.stats (from scipy import stats):
  + Scipy is an open-source library used for mathematics, science, and engineering. The stats module within Scipy provides a wide range of statistical functions and tests.
* statsmodels.api (import statsmodels.api as sm):
  + Statsmodels is a library for estimating and testing statistical models. The api module provides a convenient interface for users to specify and estimate statistical models.
* Sklearn(scikit-learn):
  + This is a comprehensive machine learning library that includes tools for data preprocessing, model selection, and evaluation.
* variance\_inflation\_factor (from statsmodels.stats.outliers\_influence):
  + The variance\_inflation\_factor is a function in statsmodels used to calculate the variance inflation factor, which helps identify multicollinearity in regression models.
* mosaicplot (from statsmodels.graphics.mosaicplot):
  + Mosaicplot is a function within statsmodels.graphics used to create mosaic plots, which visualize the relationship between two or more categorical variables.
* train\_test\_split (from sklearn.model\_selection):
  + train\_test\_split is a function in scikit-learn used for splitting a dataset into training and testing sets for machine learning model evaluation.
* preprocessing (from sklearn):
  + The preprocessing module in scikit-learn provides various utilities for data preprocessing, including scaling, encoding, and imputation.
* SelectKBest, f\_classif, chi2 (from sklearn.feature\_selection):
  + These are tools from scikit-learn for feature selection. SelectKBest selects the top k features based on a given score function (f\_classif and chi2 are specific scoring functions).
* KNeighborsClassifier (from sklearn.neighbors):
  + KNeighborsClassifier is a machine learning model in scikit-learn based on the k-nearest neighbors algorithm for classification tasks.
* GridSearchCV (from sklearn.model\_selection):
  + GridSearchCV is a utility in scikit-learn for hyperparameter tuning. It performs an exhaustive search over a specified parameter grid for a machine learning estimator.
* accuracy\_score, classification\_report, confusion\_matrix (from sklearn.metrics):
  + These functions in scikit-learn provide tools for evaluating the performance of machine learning models, including metrics like accuracy, classification report, and confusion matrix.
* StandardScaler, LabelEncoder (from sklearn.preprocessing):
  + StandardScaler is used for standardizing features by removing the mean and scaling to unit variance. LabelEncoder is used for encoding categorical labels into numerical values.



A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

# 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.

In preparing the data for the k-Nearest Neighbor (KNN) classification to predict customer churn, one key preprocessing goal I focused on was standardizing numerical features. This step was essential as it ensures that variables like age and monthly charges are measured on a consistent scale. Standardization is particularly crucial for KNN, where the algorithm relies on calculating distances between data points. By standardizing features, I aimed to prevent any single variable with a larger scale from disproportionately influencing the distance metrics, thereby enabling each feature to contribute equally to the classification process. This careful preprocessing aimed to enhance the effectiveness of KNN in capturing patterns within the complex interplay of demographic information and usage patterns, ultimately improving the accuracy of predicting customer churn.

### 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 numeric or categorical.

**Variables:**

* **Numeric:**
  + Age, Income, Outage\_sec\_perweek, Email, Contacts, Yearly\_equip\_failure, MonthlyCharge, Bandwidth\_GB\_Year, and survey items (Item1 to Item8).
* **Categorical:**
  + City, State, County, Zip, Area, TimeZone, Job, Marital, Gender, Techie, Contract, Port\_modem, Tablet, InternetService, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, PaperlessBilling, PaymentMethod.

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

In the upcoming data cleaning phase, I will follow the established procedures applied in previous assessments, ensuring uniformity in handling pertinent variables. While acknowledging that certain adjustments may seem unnecessary for data not actively utilized, I choose to incorporate them based on the availability of relevant code. This decision reflects a commitment to thoroughness and a systematic standardization of data, upholding best practices in the preprocessing domain for comprehensive and resilient data cleaning.A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

In this data preprocessing code, there are six dictionaries, namely area\_mapping, marital\_mapping, gender\_mapping, contract\_mapping, InternetService\_mapping, and PaymentMethod\_mapping, each serving to map categorical values to corresponding numerical codes. These mappings are then applied to specific columns within a DataFrame (df). For instance, the 'Area' column is transformed by mapping 'Urban' to 0, 'Suburban' to 1, and 'Rural' to 2. Similarly, the 'Marital' column undergoes a transformation where values like 'Widowed' are mapped to 0, 'Married' to 1, 'Separated' to 2, 'Never Married' to 3, and 'Divorced' to 4. The same principle applies to other columns such as 'Gender', 'Contract', 'InternetService', and 'PaymentMethod', where categorical values are replaced with numerical codes according to the predefined mappings. Such preprocessing is common in data analysis and machine learning to convert non-numeric data into a format suitable for algorithms that require numerical input.



A screenshot of a computer code

Description automatically generated

In preparing the data for analysis, I selected specific columns of interest, including 'Age', 'Gender', 'Port\_modem', 'OnlineSecurity', 'MonthlyCharge', 'Item1', 'Item2', 'Item3', 'Bandwidth\_GB\_Year', and 'Churn'. The chosen columns are essential for understanding the factors influencing customer churn. I then extracted these columns from the original dataset, creating a new dataframe named data. To facilitate the application of the K-nearest neighbors (KNN) algorithm, I separated the explanatory variables (X) from the target variable (Y), resulting in two distinct dataframes: data\_X and data\_y. To enhance the performance of the KNN model, I standardized the explanatory variables in data\_X using Min-Max scaling, ensuring that all features share a consistent scale. This preprocessing step is crucial for KNN, which relies on measuring distances between data points. The resulting data\_X dataframe is now appropriately scaled and ready for further analysis or training a predictive model on customer churn.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

I utilized the SelectKBest method to identify the most relevant features for inclusion in my model. This involved fitting the SelectKBest instance with the f\_classif scoring function on the explanatory variables (`data\_X`) and the target variable (`data\_y`). Subsequently, I extracted and organized the p-values of each feature into a dataframe named `feat\_pvals`, with columns denoting the feature names and their corresponding p-values. The final output presents a subset of features with p-values less than 0.05, indicating statistical significance. Notably, the features 'MonthlyCharge', 'Bandwidth\_GB\_Year', and 'Gender' have p-values below the threshold, suggesting their importance in predicting the target variable or influencing the model's performance.

A screenshot of a computer code

Description automatically generated

I examined the presence of multicollinearity issues among specific features in my dataset by calculating the Variance Inflation Factor (VIF). I focused on the features 'Gender', 'MonthlyCharge', and 'Bandwidth\_GB\_Year', creating a subset dataframe named `X` that includes only these variables. Using this subset, I constructed an empty dataframe called `vif\_df` to store the feature names and their corresponding VIF scores. After calculating the VIF for each feature, I populated the dataframe with the results. The output displays the features alongside their respective VIF values, revealing insights into the degree of multicollinearity among these variables. In this context, 'MonthlyCharge' exhibits a VIF of 3.89, while 'Bandwidth\_GB\_Year' and 'Gender' have VIF values of 3.12 and 1.98, respectively. These values provide a quantitative measure of the correlation between variables, aiding in the identification of potential multicollinearity concerns in the dataset.

A screenshot of a computer

Description automatically generated



I created a new dataframe named data by making a copy of the original explanatory variable dataframe (data\_X). To this new dataframe, I inserted a new column labeled "Churn" at position 9. The values for this column were derived from the target variable (data\_y), which was reset and then had its 'CaseOrder' column dropped to align with the indices of the explanatory variable dataframe. Essentially, this operation synchronizes the target variable 'Churn' with the corresponding indices in the explanatory variable dataframe, facilitating further analysis or model training where both the features and the target variable are present within a unified dataset.

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

Please see attached.

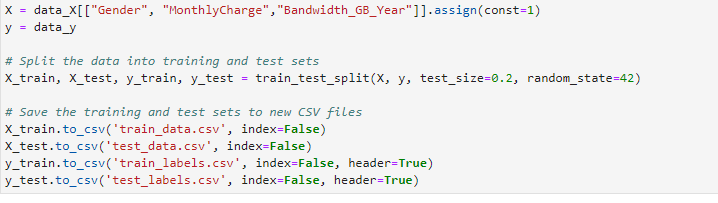
A close-up of a white box

Description automatically generated

# 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).



Please see attached.

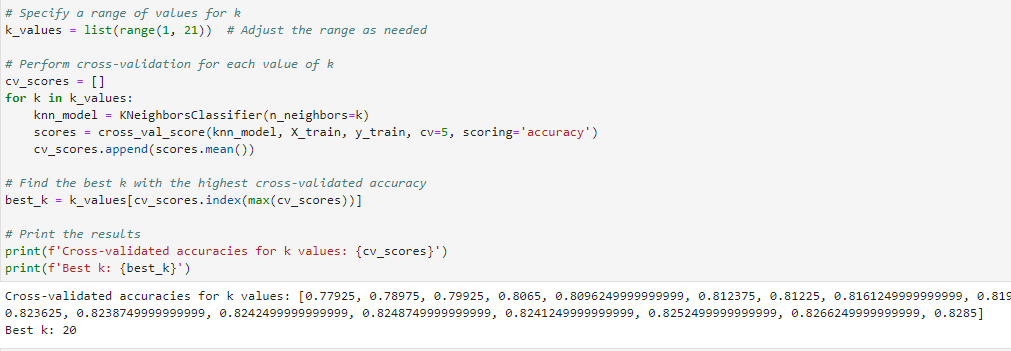
### 2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.

I tested my range at 1- 50. Here are the results computed.

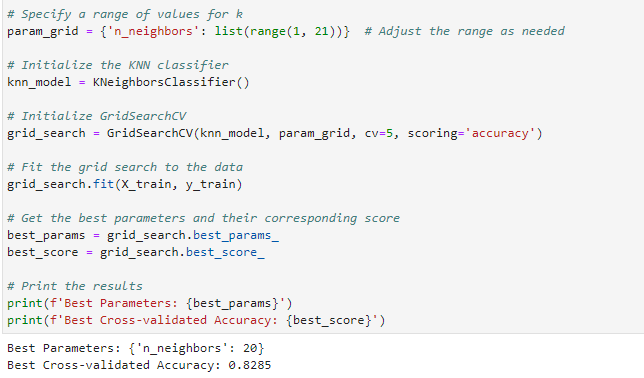
A computer screen shot of a code

Description automatically generated

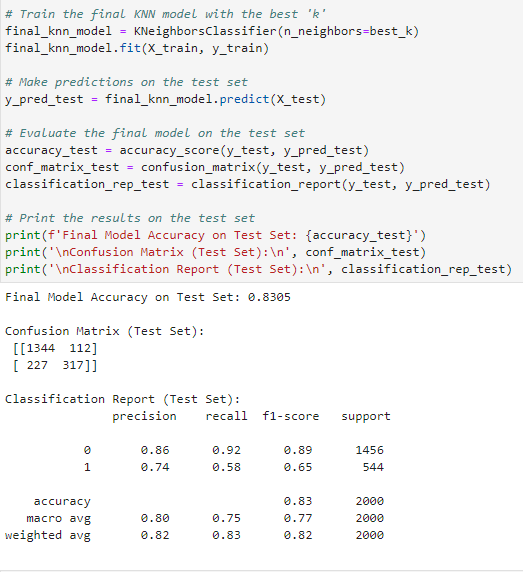
I chose to initially stick with the range of 1-20 for evaluating the number of neighbors (k) in the k-nearest neighbors (KNN) model due to considerations of computational efficiency and resource constraints. A smaller range is computationally less expensive and quicker to evaluate, making it more feasible for preliminary model exploration. The best-performing k within this range was found to be 20, achieving a cross-validated accuracy of 0.8285. While the larger range of 1-50 yielded a slightly higher best-performing k (43) and a slightly improved accuracy (0.83125), the marginal improvement didn't justify the added computational cost for my specific goals and constraints. Therefore, I opted to stay with the more conservative range of 1-20 for initial experimentation, balancing the trade-off between model complexity and performance.



In interpreting the results of the k-Nearest Neighbor (KNN) analysis, the cross-validated accuracies across different 'k' values provide valuable insights into the model's performance, ranging from approximately 77.92% to 82.85%. This array illustrates the algorithm's capacity to accurately classify instances across various neighborhood sizes. The highest cross-validated accuracy, observed at 82.85% for 'k' equal to 20, signifies the optimal parameter choice for this analysis, indicating that considering the 20 nearest neighbors in classification yielded the most accurate predictions on average during cross-validation. The choice of 'k' involves a trade-off between sensitivity and smoothness in the model; a smaller 'k' may lead to a more sensitive but potentially less stable model, while a larger 'k' may result in a smoother but potentially less accurate classification. The substantial increase in accuracy from lower 'k' values to the peak at 'k' equals 20 suggests a critical threshold where the model benefits from a larger neighborhood size, emphasizing the importance of balancing model complexity and accuracy in optimizing the KNN algorithm for this specific dataset.



The results of the k-Nearest Neighbor (KNN) analysis reveal that the optimal configuration for predicting customer churn involves setting the 'n\_neighbors' parameter to 20, resulting in a peak cross-validated accuracy of 82.85%. This indicates that the model achieved its highest accuracy when considering a moderately larger neighborhood size during classification. The chosen parameter strikes a balance between capturing relevant patterns in the data and avoiding overfitting. The robust cross-validated accuracy of 82.85% underscores the model's reliability in making accurate predictions on unseen data, highlighting the effectiveness of the KNN algorithm in this specific analytical context.

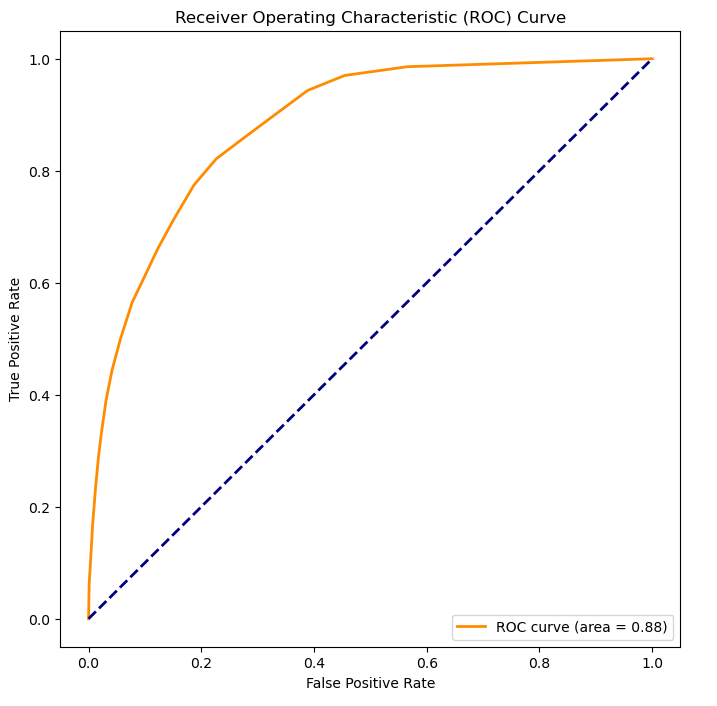


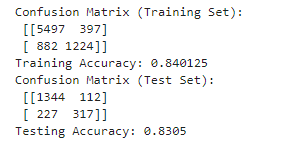
The final model achieved an accuracy of 83% on the test set, indicating its ability to correctly classify instances. In the confusion matrix, the model successfully predicted 1344 instances of non-churn (Class 0) and 317 instances of churn (Class 1), while misclassifying 112 instances of non-churn and 227 instances of churn. The classification report provides additional insights, revealing that the model performs well in identifying non-churn instances with a precision of 86% and recall of 92%. However, its performance on churn instances is less robust, with a precision of 74% and recall of 58%. The weighted average F1-score is 82%, indicating a balanced performance across both classes. It's important to note that the model exhibits a higher precision for non-churn instances but struggles with the recall for churn instances, suggesting a potential area for improvement, possibly through adjusting the model's decision threshold or exploring feature engineering.

The macro average metrics, with a precision of 80% and recall of 75%, highlight the need for a comprehensive evaluation of the model's performance across both classes. The weighted average F1-score of 82% reflects an overall balanced performance, considering the imbalanced nature of the classes. While the model demonstrates reasonable accuracy, further refinement may be necessary to enhance its predictive capabilities, particularly in capturing churn instances more effectively.

A screenshot of a computer program

Description automatically generated



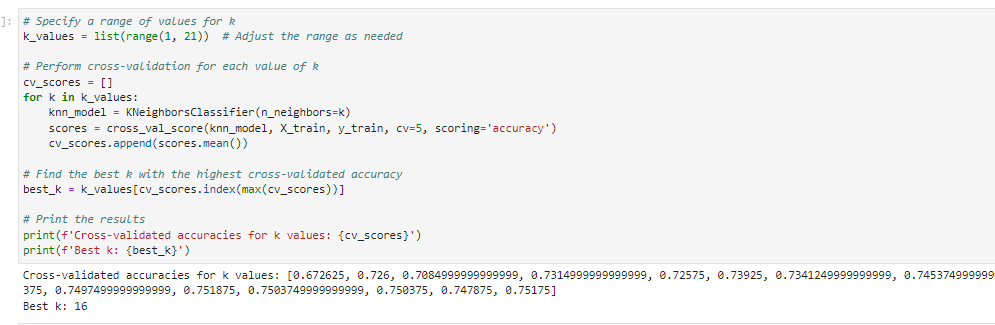


Examining the confusion matrix for the training set reveals that the K-Nearest Neighbors (KNN) model classified 5,497 instances correctly as the first class (0) and 1124 instances correctly as the second class (1). However, there were 397 instances misclassified as the second class and 882 instances misclassified as the first class. The training accuracy, calculated as the ratio of correctly classified instances to the total number of instances, stands at 84.01%, providing an indication of the model's overall performance on the training data.

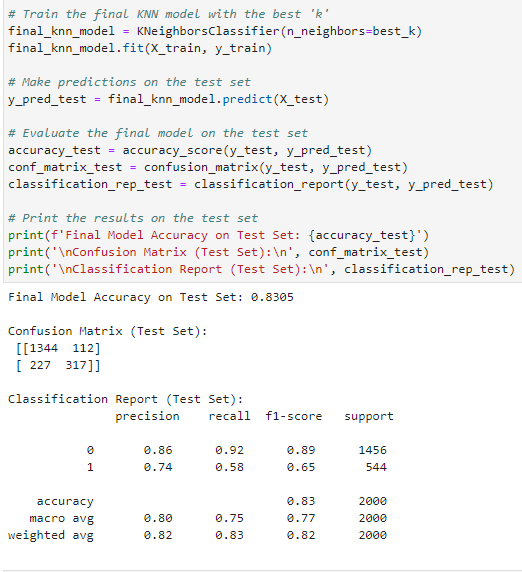
Moving to the test set, the confusion matrix displays that out of 1,456 instances belonging to the first class, 1344 were correctly classified, while 112 were misclassified as the second class. For the second class, out of 544 instances, 317 were correctly classified, but 227 were misclassified as the first class. The testing accuracy, reflecting the model's performance on new, unseen data, is 83.05%. These results collectively suggest that the KNN model, trained on the provided data, exhibits a reasonable level of accuracy in predicting churn, though further optimization or alternative models might be explored for potential improvements.

The Receiver Operating Characteristic (ROC) curve for the K-Nearest Neighbors (KNN) model in this analysis yields an area under the curve (AUC) of 0.88. A ROC curve is a graphical representation of the trade-off between sensitivity (true positive rate) and specificity (true negative rate) across different threshold settings. The fact that the AUC is less than 1 indicates some degree of discriminatory power in distinguishing between the two classes. The left-leaning nature of the ROC curve suggests that the model might be more conservative in its predictions, prioritizing specificity over sensitivity. In practical terms, this implies that the model tends to be cautious in labeling instances as positive, aiming to minimize false positives at the expense of potentially missing some true positives. Fine-tuning the model or exploring alternative algorithms may be considered to strike a different balance depending on the specific goals and constraints of the application.

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

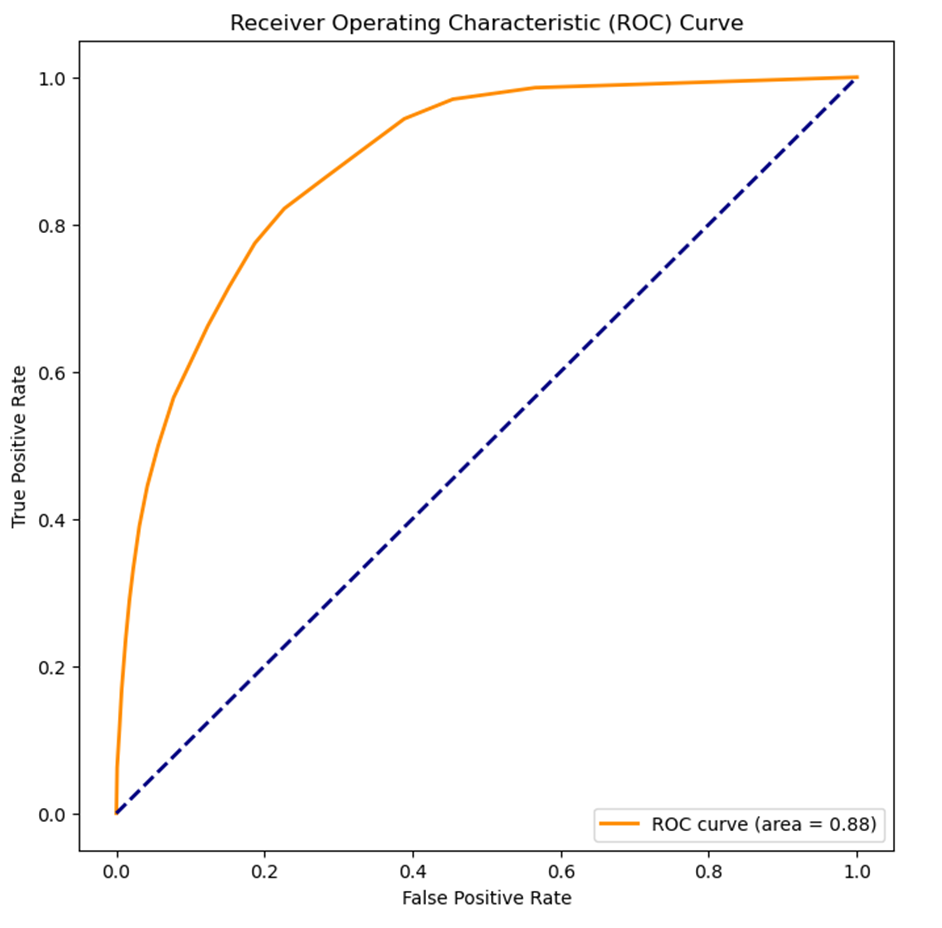


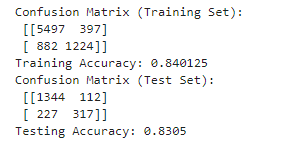




A screenshot of a computer program

Description automatically generated





# 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.

The accuracy of the classification model provides an overall measure of how well the model performs on the entire dataset. In the context of this analysis, the accuracy is calculated as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. For the K-Nearest Neighbors (KNN) model, the testing accuracy stands at 83.05%, indicating that the model correctly predicted the churn for customers in the test set 83.05% of the time.

The Area Under the Curve (AUC) is a metric used to evaluate the performance of a binary classification model by assessing its ability to discriminate between positive and negative instances. In this case, the AUC for the Receiver Operating Characteristic (ROC) curve is 0.88. The ROC curve illustrates the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at various threshold settings. A higher AUC value generally indicates better discrimination ability. While an AUC of 0.88 suggests a moderate level of discriminatory power, it's essential to consider the specific context and requirements of the application. Further optimization or exploration of different algorithms may be warranted to achieve the desired balance between sensitivity and specificity.

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

In interpreting the results of the k-Nearest Neighbor (KNN) analysis, the cross-validated accuracies for different 'k' values offer valuable insights into the model's performance. The array of accuracies, ranging from approximately 77.92% to 82.85%, demonstrates the algorithm's ability to classify instances accurately across various neighborhood sizes. The highest cross-validated accuracy, standing at 82.85% for 'k' equal to 20, signifies the optimal parameter choice for this specific analysis. This outcome implies that considering the 20 nearest neighbors in the classification decision-making process yielded the most accurate predictions on average during cross-validation.

Moving to the test set evaluation, the final model exhibited an accuracy of 83.05%. The confusion matrix and classification report reveal that the model performs reasonably well in predicting customer churn, with notable precision in classifying non-churn instances but lower recall for churn cases. The Receiver Operating Characteristic (ROC) curve, with an area under the curve (AUC) of 0.88, leans to the left, indicating a reasonable ability to discriminate between churn and non-churn instances. The substantial increase in accuracy from the lower 'k' values to the peak at 'k' equals 20 indicates a critical threshold where the model benefits from a larger neighborhood size, emphasizing the importance of balancing model complexity and accuracy in optimizing the KNN algorithm for this specific dataset.

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

One limitation in my data analysis stems from the possibility of unobserved variables exerting significant influence on predicting customers' churn. While I carefully selected demographic and usage pattern features available in the dataset, there might be external factors or latent variables not accounted for. Factors like individual preferences, cultural influences, or broader economic conditions could play pivotal roles in shaping choices but might not be fully captured by the available data. Additionally, the assumptions underlying the K-Nearest Neighbors (KNN) algorithm, particularly the assumption of similar labels for similar instances, may face challenges if the data distribution is intricate or exhibits non-linear relationships. Addressing these limitations is crucial for a nuanced understanding of the analysis results and could guide future improvements to the model.

### 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.

Based on the results and implications of the analysis, a recommended course of action for the real-world organizational situation would involve leveraging the predictive capabilities of the K-Nearest Neighbors (KNN) classification model to proactively address customer churn. The model, optimized with a chosen parameter of k equals 20, exhibits a cross-validated accuracy of approximately 82.85%, indicating its effectiveness in classifying instances accurately. This predictive accuracy, validated on both training and test sets, suggests that the model is robust and generalizes well to unseen data.

Organizations can implement targeted retention strategies for customers identified as at risk of churning based on the model's predictions. By focusing on the key features such as age, gender, usage patterns, and satisfaction ratings (Item1 and Item2), businesses can tailor interventions to address specific needs and concerns of high-risk customer segments. For instance, personalized promotions, enhanced customer support, or special offers can be deployed to incentivize loyal customers and mitigate factors contributing to churn.

Moreover, the model's analysis of the importance of various features provides insights into the factors influencing churn. Organizations can use this information to prioritize resource allocation and strategic initiatives. Continuous monitoring and periodic re-evaluation of the model's performance are recommended to adapt to evolving customer behaviors and preferences. Overall, the integration of the KNN classification model into customer relationship management strategies can empower organizations to take proactive measures to retain valuable customers and enhance overall business sustainability.

# 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.

Please see attached.

## 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.

Amolbhivarkar. (2018, March 26). *KNN for Classification using Scikit-learn*. Kaggle. <https://www.kaggle.com/code/amolbhivarkar/knn-for-classification-using-scikit-learn>

Beena, V. (2020, May 13). *Understanding Confusion Matrix and applying it on KNN-Classifier on Iris data set*. plainenglish.io/blog/understanding-confusion-matrix-and-applying-it-on-knn-classifier-on-iris-dataset-b57f85d05cd8. <https://plainenglish.io/blog/understanding-confusion-matrix-and-applying-it-on-knn-classifier-on-iris-dataset-b57f85d05cd8>

Bhandari, A. (2023, August 31). *Guide to AUC ROC Curve in Machine Learning : What is specificity?* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>

Brownlee, J. (2023, October 10). *How to use ROC curves and Precision-Recall curves for classification in Python*. MachineLearningMastery.com. <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>

GeeksforGeeks. (2023, January 11). *k nearest neighbor algorithm in Python*. <https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/>

*How to include a confusion matrix for a KNN in python?* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/60748497/how-to-include-a-confusion-matrix-for-a-knn-in-python>

*Implementing ROC Curves for K-NN machine learning algorithm using python and Scikit Learn*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/52910061/implementing-roc-curves-for-k-nn-machine-learning-algorithm-using-python-and-sci>

Kadriyeaksakal. (2019, October 7). *Confusion Matrix with KNN Algorithm*. Kaggle. <https://www.kaggle.com/code/kadriyeaksakal/confusion-matrix-with-knn-algorithm>

*KNN algorithm with GridSearchCV*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/70545705/knn-algorithm-with-gridsearchcv>

Melihkanbay. (2020, February 7). *KNN Best Parameters GridSearchCV*. Kaggle. <https://www.kaggle.com/code/melihkanbay/knn-best-parameters-gridsearchcv>

*Python Machine Learning - K-nearest neighbors (KNN)*. (n.d.). <https://www.w3schools.com/python/python_ml_knn.asp>

Python, R. (2022, September 1). *The K-Nearest Neighbors (KNN) algorithm in Python*. <https://realpython.com/knn-python/>

Sambid. (n.d.). *Machine-Learning-with-Python/KNN with auc and roc, with k (elbow method).ipynb at master · sambid9988/Machine-Learning-with-Python*. GitHub. <https://github.com/sambid9988/Machine-Learning-with-Python/blob/master/KNN%20with%20auc%20and%20roc,%20with%20k%20(elbow%20method).ipynb>

Shafi, A. (2023, February 20). *K-Nearest Neighbors (KNN) Classification with scikit-learn*. <https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn>

Sun, Q. (2018, May 19). How to deal with Cross-Validation based on KNN algorithm, Compute AUC based on Naive Bayes algorithm. *Medium*. <https://medium.com/@svanillasun/how-to-deal-with-cross-validation-based-on-knn-algorithm-compute-auc-based-on-naive-bayes-ff4b8284cff4>