Task 2 D209

WGU | D209

Performance Assessment

Narcisse, Laurie

2023

Contents

[Part I: Research Question 2](#_Toc153799018)

[A.  Describe the purpose of this data mining report by doing the following: 2](#_Toc153799019)

[1.  Propose one question relevant to a real-world organizational situation that you will answer using one of the following prediction methods: 2](#_Toc153799020)

[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](#_Toc153799021)

[Part II: Method Justification 3](#_Toc153799022)

[B.  Explain the reasons for your chosen prediction method from part A1 by doing the following: 3](#_Toc153799023)

[1.  Explain how the prediction method you chose analyzes the selected data set. Include expected outcomes. 3](#_Toc153799024)

[2.  Summarize one assumption of the chosen prediction method. 3](#_Toc153799025)

[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](#_Toc153799026)

[Part III: Data Preparation 7](#_Toc153799027)

[C.  Perform data preparation for the chosen data set by doing the following: 7](#_Toc153799028)

[1.  Describe one data preprocessing goal relevant to the prediction method from part A1. 7](#_Toc153799029)

[2.  Identify the initial data set variables that you will use to perform the analysis for the prediction question from part A1, and group *each* variable as numeric or categorical. 7](#_Toc153799030)

[3.  Explain the steps used to prepare the data for the analysis. Identify the code segment for each step. 8](#_Toc153799031)

[4.  Provide a copy of the cleaned data set. 10](#_Toc153799032)

[Part IV: Analysis 11](#_Toc153799033)

[D.  Perform the data analysis and report on the results by doing the following: 11](#_Toc153799034)

[1.  Split the data into training and test data sets and provide the file(s). 11](#_Toc153799035)

[2.  Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed. 11](#_Toc153799036)

[3.  Provide the code used to perform the prediction analysis from part D2. 16](#_Toc153799037)

[Part V: Data Summary and Implications 18](#_Toc153799038)

[E.  Summarize your data analysis by doing the following: 18](#_Toc153799039)

[1.  Explain the accuracy and the mean squared error (MSE) of your prediction model. 18](#_Toc153799040)

[2.  Discuss the results and implications of your prediction analysis. 19](#_Toc153799041)

[3.  Discuss one limitation of your data analysis. 20](#_Toc153799042)

[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. 21](#_Toc153799043)

[Part VI: Demonstration 21](#_Toc153799044)

[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. 21](#_Toc153799045)

[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. 21](#_Toc153799046)

# **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 prediction methods:

Can we predict customer churn based on demographic data, customer behavior, and responses to the satisfaction survey?

### 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.

In pursuit of constructing a predictive model, our primary objective is to identify potential churners within the customer base. This endeavor aims to equip the organization with a robust tool for recognizing customers who are at risk of discontinuing their services. By leveraging predictive analytics, we intend to discern patterns and factors that contribute to customer churn. This proactive approach enables the implementation of targeted retention strategies, allowing the organization to take preemptive measures in retaining valuable customers.

The significance of this predictive model lies in its potential to foresee customer churn before it occurs. Through data-driven insights, we seek to understand the key indicators and behaviors that precede churn events. By harnessing the power of decision trees, random forests, or advanced regression techniques, we aim to create a model capable of making accurate predictions based on historical customer data.

The overarching goal is to empower the organization with actionable intelligence, enabling timely and strategic interventions. This predictive model not only serves as a preventative measure against customer attrition but also forms the foundation for a proactive customer retention strategy. Through the identification of potential churners, the organization can tailor retention efforts, providing personalized incentives, improved customer experiences, or targeted communication to mitigate the likelihood of churn. Ultimately, the purpose of this data mining report is to furnish the organization with valuable insights that pave the way for effective and targeted customer retention initiatives.

# **Part II: Method Justification**

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

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

In the context of our scenario, decision trees emerge as a fitting choice for the predictive modeling task. Their suitability stems from their ability to adeptly handle both numeric and categorical data, rendering them versatile and accommodating to the diverse nature of our dataset. Decision trees excel in situations where the dataset encompasses a mix of different data types, allowing us to navigate through the complexities of customer information that includes both numerical and categorical attributes.

One notable strength of decision trees is their innate interpretability. The generated tree structure offers a clear and intuitive representation of the decision-making process. This interpretability is crucial for understanding the factors that contribute to churn predictions. By examining the branches and nodes of the tree, we can discern the specific variables and their interactions that play a pivotal role in determining whether a customer is likely to churn.

Moreover, decision trees are well-suited for capturing non-linear relationships within the data. This flexibility is invaluable in scenarios where the underlying patterns may not follow linear trends. As a result, the expected outcome of employing decision trees in our analysis is the creation of a comprehensible tree structure. This structure will serve as a visual and interpretable guide, unveiling the influential factors that drive our churn predictions. Overall, the versatility, interpretability, and non-linear capturing capabilities of decision trees make them a valuable tool for uncovering insights within our dataset.

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

Decision trees operate under the assumption that the features within the dataset are meaningful and can be efficiently divided to make accurate predictions. However, it's essential to note their sensitivity to outliers, as these extreme values can disproportionately impact the splitting process. The influence of outliers can potentially lead to suboptimal splits and affect the overall performance of the decision tree model. It's crucial to consider the potential impact of outliers when employing decision trees and, if necessary, implement preprocessing techniques to mitigate their effects and enhance the robustness of the model.

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

In my analysis, I've chosen a set of Python libraries to support various stages of the decision tree analysis. Here's how each item on the list contributes to the overall process:

* pandas (pd):
  + Personally, I rely on pandas to manage and manipulate the dataset efficiently. It provides essential data structures like DataFrames, facilitating data organization, exploration, and preprocessing.
* numpy (np):
  + For numerical operations and array manipulations, I turn to NumPy. It helps handle numerical data effectively, which is crucial when working with features and labels.
* matplotlib.pyplot (plt):
  + Matplotlib's pyplot aids in creating diverse plots. Visualization is key to understanding data distributions, relationships, and model performance, making it an indispensable tool in my analysis.
* seaborn (sns):
  + Building on Matplotlib, seaborn enhances the visual appeal of statistical graphics. It assists in creating informative and aesthetically pleasing visualizations, contributing to a deeper understanding of data patterns during exploratory data analysis.
* scipy.stats:
  + Scipy's stats module complements my analysis with statistical functions. It supports hypothesis testing, statistical measures calculation, and exploration of variable distributions, providing deeper insights into the dataset.
* sklearn.model\_selection.train\_test\_split:
  + Personally, I use this function from scikit-learn to split the dataset into training and testing sets. It supports the model training process by providing separate datasets for training and evaluation.
* sklearn.preprocessing.LabelEncoder:
  + LabelEncoder is crucial for converting categorical variables into numerical format. This preprocessing step ensures the model can interpret and use categorical information effectively.
* sklearn.tree.DecisionTreeClassifier, sklearn.tree.export\_text:
  + These scikit-learn components form the core of my decision tree analysis. DecisionTreeClassifier builds the model, while export\_text allows for visualizing the decision tree rules, aiding in the understanding of the model's decision-making process.
* sklearn.metrics.accuracy\_score, sklearn.metrics.mean\_squared\_error, sklearn.metrics.confusion\_matrix:
  + Scikit-learn's metrics help evaluate the model's performance comprehensively. accuracy\_score measures overall correctness, mean\_squared\_error quantifies average squared differences, and confusion\_matrix offers insights into the model's performance across different classes.
* sklearn.model\_selection.GridSearchCV:
  + GridSearchCV automates hyperparameter tuning, optimizing the decision tree model's parameters and enhancing its predictive performance.
* sklearn.metrics.roc\_auc\_score, sklearn.metrics.roc\_curve, sklearn.metrics.auc:
  + These metrics from scikit-learn are vital for evaluating the model's discrimination ability. ROC analysis and AUC summarize the trade-off between true positive and false positive rates.
* sklearn.ensemble.AdaBoostClassifier:
  + Personally, I leverage AdaBoost to enhance the decision tree model's performance through ensemble learning. This is particularly useful in scenarios with imbalanced data or complex relationships.
* sklearn.metrics.classification\_report:
  + Classification\_report provides a detailed overview of classification metrics, offering valuable insights into the model's performance across different classes.

A screenshot of a computer program

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 prediction method from part A1.

One key objective in preprocessing data for decision trees involves personally managing missing values and encoding categorical variables to optimize the subsequent analysis. It is crucial to address any missing values systematically, employing techniques like imputation to replace gaps in the data with suitable estimates derived from the available information. Additionally, as decision tree algorithms inherently operate with numerical data, I need to personally ensure that categorical variables are transformed into a numerical format through encoding. This transformation is vital to enable the decision tree model to accurately interpret and utilize the information embedded in categorical features. By prioritizing the handling of missing values and encoding of categorical variables, I lay the groundwork for a robust and accurate decision tree analysis, enabling the model to extract meaningful insights from diverse datasets in a manner aligned with my specific goals and requirements.

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

* Numeric Variables:
  + Outage\_sec\_perweek, Email, Contacts, Yearly\_equip\_failure, Age, Income, MonthlyCharge, Bandwidth\_GB\_Year, Item1 to Item8.
* Categorical Variables:
  + 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 the steps used to prepare the data for the analysis. Identify the code segment for each step.

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

A white rectangular sign with black text

Description automatically generated



A screenshot of a computer

Description automatically generated

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

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).

A screen shot of a computer code

Description automatically generated

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

To construct an effective decision tree classification model using the DecisionTreeClassifier, I need to optimize its parameters, specifically focusing on the max\_depth and min\_samples\_leaf fields. The max\_depth parameter determines the maximum allowable depth of the tree, influencing its complexity; however, a deeper tree may lead to overfitting. On the other hand, min\_samples\_leaf governs the minimum number of samples required for a node to be considered a leaf, impacting how the tree splits its nodes.

To achieve this optimization, I'll start by creating an initial instance of the DecisionTreeClassifier and defining a range of potential values for max\_depth and min\_samples\_leaf. Employing GridSearch, I'll systematically evaluate the performance of the initial decision tree model across various parameter combinations. The GridSearch will identify the optimal set of parameters that yield the best model performance. Subsequently, I'll display the accuracy and AUC score of this tuned model, providing a baseline for further refinement.

After the GridSearch completes the hyperparameter tuning, I will instantiate a new DecisionTreeClassifier using the determined optimal parameters. To enhance the resulting decision tree, I plan to leverage Adaptive Boosting (AdaBoosting). However, AdaBoosting requires an initial decision tree, making the thorough hyperparameter tuning crucial for creating a robust foundation. This meticulous process ensures that the subsequent AdaBoosting step builds upon the most effective decision tree, ultimately leading to an improved and highly accurate classification model.

A screenshot of a computer code

Description automatically generated

* Test set accuracy of the best decision tree: 0.84
  + This means that, when applying the trained decision tree model to a previously unseen test set, the model correctly predicted the target variable for approximately 84% of the instances. In other words, 84% of the predictions made by the model on the test set were accurate.
* Test set ROC AUC score: 0.902
  + The ROC AUC score, which stands for Receiver Operating Characteristic Area Under the Curve, is a metric that evaluates the model's ability to distinguish between the positive and negative classes. In this case, the ROC AUC score is 0.902, indicating that the model demonstrates strong discriminatory power. The closer the ROC AUC score is to 1, the better the model is at distinguishing between the two classes. A score of 0.902 suggests that the model has a high true positive rate and a low false positive rate.

In summary, these results indicate that the trained decision tree model performs well on the test set, with an accuracy of 84% and a robust ability to discriminate between the classes, as reflected in the high ROC AUC score of 0.902.

A screenshot of a computer

Description automatically generated

I found that the best-performing model for my dataset is a Decision Tree Classifier with the following hyperparameters: a maximum depth of 6, a minimum number of samples per leaf set to 0.04, and a random state of 42. When applying this model to a previously unseen test set, I achieved a test set accuracy of 84%, indicating that the model accurately predicted the target variable for 84% of the instances. Additionally, the ROC AUC score, which measures the model's ability to discriminate between classes, was 0.902. This high ROC AUC score suggests that the model has a strong capability to distinguish between positive and negative classes, making it an effective choice for my classification task.

A screenshot of a computer program

Description automatically generated

I discovered that the optimal hyperparameters for my AdaBoost model are a learning rate of 1.0 and 180 estimators. After fine-tuning these parameters using a grid search, the combination of a learning rate of 1.0 and 180 estimators yielded the best performance for my specific dataset. This suggests that, for my AdaBoost model, a moderate learning rate and a substantial number of estimators contribute to its effectiveness in making predictions. These hyperparameters were identified as the most effective for maximizing the model's performance on my task.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer error

Description automatically generated

In my final decision tree model, the test set accuracy is 0.82, indicating that the model correctly predicted the target variable for 82% of the instances in the previously unseen test set. This suggests a reasonably good overall performance of the decision tree on the test data.

The Area Under the Curve (AUC) score, a metric evaluating the model's ability to distinguish between positive and negative classes, is 0.89. This high AUC score suggests that the decision tree model demonstrates strong discriminatory power, with a good balance between true positives and false positives.

The confusion matrix provides a more detailed view of the model's performance. It shows that out of 1470 instances with the true label 0 , the model correctly predicted 1326, but misclassified 144 instances. For the true label 1, the model correctly predicted 323 instances, but misclassified 207 instances. This gives insights into the model's performance across different classes.

The mean squared error (MSE) of the model is 0.1755, providing an overall measure of the average squared difference between predicted and actual values. The root mean squared error (RMSE) is 0.42, offering a more interpretable measure of the average magnitude of the prediction errors.

The classification report breaks down precision, recall, and F1-score for both classes (0 and 1). Precision is the ratio of correctly predicted positive observations to the total predicted positives, recall is the ratio of correctly predicted positive observations to all the actual positives, and F1-score is the weighted average of precision and recall. This report gives a comprehensive view of the model's performance across different evaluation metrics for each class.

In summary, the model exhibits good accuracy and AUC score, but the detailed analysis provided by the confusion matrix and classification report sheds light on its performance across individual classes and various metrics.

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

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer error

Description automatically generated

# **Part V: Data Summary and Implications**

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

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

In assessing the performance of my final decision tree model, I observe that the accuracy on the test set stands at 82%. This implies that the model made correct predictions for approximately 82% of the instances within the test data. Accuracy serves as a straightforward metric for overall correctness, although its interpretation benefits from consideration of other evaluation metrics, especially in scenarios involving imbalanced datasets.

Moving on to the mean squared error (MSE), I find that the model exhibits an MSE of 0.1755. MSE provides an insight into the average squared differences between the actual and predicted values. A lower MSE is indicative of a better fit of the model to the data. In the context of my model, an MSE of 0.1755 suggests that, on average, the squared discrepancies between the actual and predicted values are relatively small.

Additionally, the root mean squared error (RMSE) is derived from the MSE, resulting in a value of 0.42. This metric, in the same units as the target variable, represents the average magnitude of the prediction errors. A smaller RMSE signifies better overall model performance, indicating that the model's predictions are, on average, close to the actual values.

In conclusion, my decision tree model demonstrates a satisfactory accuracy of 82%, and the MSE and RMSE values suggest that the model's predictions exhibit a relatively small average squared difference from the actual values. To gain a more comprehensive understanding of the model's performance, I consider additional metrics such as the AUC score, confusion matrix, and classification report, particularly in the context of dealing with imbalanced datasets.

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

A screenshot of a computer program

Description automatically generated

A graph of a curve

Description automatically generated

Upon reviewing the outcomes of my prediction analysis, the final decision tree model showcases a test set accuracy of 82%. This indicates a commendable ability to accurately predict the target variable for a substantial portion of instances within the test dataset, reflecting the model's capacity to generalize effectively to new, unseen data.

The Area Under the Curve (AUC) score, a critical metric for assessing discriminatory power, stands at 0.89. This high AUC score underscores the model's effectiveness in distinguishing between positive and negative classes. Notably, the ROC curve visually reflects this discrimination, leaning convincingly to the left. Such a leaning curve suggests a robust true positive rate, emphasizing the model's ability to correctly identify instances of the positive class.

Upon delving into the confusion matrix, a closer examination of the model's performance emerges. The matrix reveals a notable number of true negatives (1326) and true positives (323). However, it also highlights areas for improvement, with 144 false positives and 207 false negatives. These elements offer insights into specific instances where the model can be refined to enhance precision and recall.

Considering mean squared error (MSE) and root mean squared error (RMSE), the model demonstrates a small average magnitude of prediction errors, with an MSE of 0.1755 and an RMSE of 0.42. This suggests a solid overall fit of the model to the data, contributing to the reliability of its predictions.

The classification report further dissects the model's performance, providing detailed metrics such as precision, recall, and F1-score for each class. With a weighted average F1-score of 0.82, the model strikes a balance between precision and recall, particularly beneficial in scenarios with class imbalances.

In summary, the final decision tree model presents encouraging results, showcasing a favorable blend of accuracy, discriminatory power, and effective generalization to unseen data. The left-leaning ROC curve and high AUC score affirm the model's robustness in correctly identifying positive instances. The nuanced insights from the confusion matrix and classification report provide actionable areas for refinement, contributing to ongoing efforts to optimize the model's performance.

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

One limitation I encountered in the data analysis is the potential impact of imbalanced classes on the model's performance. The dataset exhibits a notable difference in the number of instances between the two classes, with significantly more instances of the negative class than the positive class. This class imbalance can influence the model's learning process, potentially leading to biases and suboptimal performance.

In situations where one class dominates the dataset, the model might develop a bias towards predicting the majority class, resulting in high accuracy for that class but poor performance for the minority class. In my analysis, despite achieving a respectable overall accuracy of 82%, it's crucial to consider how well the model performs for both classes individually.

The imbalanced nature of the data may have implications for metrics like precision, recall, and F1-score, particularly for the minority class. These metrics can be sensitive to imbalances, potentially providing an incomplete picture of the model's effectiveness in capturing patterns for the less prevalent class.

To address this limitation, techniques such as resampling, using different evaluation metrics, or exploring advanced algorithms designed to handle imbalanced datasets may be worth considering. Acknowledging and mitigating the impact of imbalanced classes is vital for ensuring a more comprehensive and equitable assessment of the model's capabilities in real-world scenarios.

### 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 findings from my analysis of the research question—“Can we predict customer churn based on demographic data, customer behavior, and responses to the satisfaction survey?”—I recommend several actions to enhance our approach to predicting and managing customer churn.

Firstly, addressing the issue of imbalanced classes is crucial. The dataset's significant class imbalance may bias the model towards the majority class, affecting its ability to predict churn accurately. Implementing techniques such as oversampling, undersampling, or synthetic data generation will help balance class distributions and improve the model's performance for both customer segments.

In addition, refining the model parameters is essential for optimal predictive performance. Fine-tuning hyperparameters, such as tree depth, minimum samples per leaf, and learning rate, through methods like grid search or randomized search, can enhance the model's ability to capture patterns in customer behavior and demographics.

In conclusion, implementing these recommendations will contribute to a more effective and accurate approach to predicting customer churn. By continuously refining the model, leveraging customer feedback, and fostering collaboration across departments, we can develop targeted retention strategies and maintain a proactive stance in managing customer relationships.

# **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.

GeeksforGeeks. (2019, June 30). Python Mean squared error. <https://www.geeksforgeeks.org/python-mean-squared-error/>

GeeksforGeeks. (2023, December 7). Python  Decision tree implementation. <https://www.geeksforgeeks.org/decision-tree-implementation-python/>

Mean Squared error in Python. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/39064684/mean-squared-error-in-python>

Navlani, A. (2023, February 23). Decision Tree Classification in Python Tutorial. <https://www.datacamp.com/tutorial/decision-tree-classification-python>

Prashant. (2020, March 13). Decision-Tree Classifier tutorial. Kaggle. <https://www.kaggle.com/code/prashant111/decision-tree-classifier-tutorial>

Python Machine Learning - AUC - ROC Curve. (n.d.). <https://www.w3schools.com/python/python_ml_auc_roc.asp>

Python Machine Learning Decision Tree. (n.d.). <https://www.w3schools.com/python/python_ml_decision_tree.asp>

Sklearn.metrics.auc. (n.d.). Scikit-learn. <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html>

sklearn.metrics.mean\_squared\_error. (n.d.). Scikit-learn. <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html>

Which is the correct way to calculate AUC with scikit-learn? (n.d.). Stack Overflow. <https://stackoverflow.com/questions/66397641/which-is-the-correct-way-to-calculate-auc-with-scikit-learn>