Sai Ganesh Pendela

ENPM808L ANALYTICS FOR DECISION SUPPORT
MIDTERM EXAM
FALL 2023

This is a take-home mid-term exam. All answers are to be provided individually by all students. No external help is allowed. Only course materials (book, homework, slides, and these exam materials) may be used as reference. By submitting this exam, you are agreeing to these conditions.

This exam will be posted in the "Midterm" assignment entry. Download a copy and enter your answers in this document.

Enter your name in the header so it appears on all pages.

Please preserve starting each question on a new page. Answers can span pages, it just helps when reading them to have them start on new pages (thanks).

Place answers for each section of a question immediately after that section. For example

1) Answer…
    a. Part 1…

<start here> Place your answer to part 1A here.

    b. Part 2…

<start here> Place your answer to part 1b here.

Complete this exam and convert the MS-Word document to a pdf file named as LastName_FirstName_Midterm_ENPM808L_20231013.pdf and submit the PDF to the assignment "Midterm" by 11:59 pm on 13 October 2023.

Keep your MS-Word document in case there are issues in conversion of the PDF.

Each of the five (5) questions will be graded A-F as noted in the syllabus.
Thus, a total of 20 points (5*4 for all A answers) is possible.

Sai Ganesh Pendela

1. Given the Midterm-IG-example data in the XLSX file, answer the following questions:
    a. Sheet1 computes IG from entropy (yellow), variance (green) of the income variable and from the variance (blue) of the write-off variable value itself. Which data split sets would be chosen from each method {IGe, IGv(income), & IGv(write-off)} and why are they the same or different? Be complete in your answer by defining the reason in terms of the algorithm and the data.

A. The choice of data split sets for the IGe (Information Gain with respect to the target variable) and IGv(income) (Information Gain with respect to income) methods aligns with higher income values, such as "Income < 60K" or "Income < 70K," as their values increase with income. Thus, they tend to prefer the same data split sets. However, IGv(write-off) values, based on information gain related to write-off, remain relatively small and do not consistently increase with income thresholds. Consequently, the data split sets favored by IGv(write-off) might differ, and depends on the importance of the write-off feature within the decision tree model and the specific use case.

    b. Sheet2 computes IG from a logical value of 1 or 0 for write-off instead of the actual write-off value. Does this change the selections made in 1a above, and why or why not?

A. Even if Sheet2 computes Information Gain (IG) based on a logical value of 1 or 0 for write-off instead of the actual write-off values, the IGe (Information Gain with respect to the target variable),IGv(income) (Information Gain with respect to income) and IGv(write-off) follow the same trends as they are in sheet1, so there may be no chnages in the selection made in 1a. But in general, there may be reduction of information content when treating write-off as a binary variable (1 indicating write-off, 0 indicating no write-off). The selections may differ because the simplified binary representation of write-off may not capture the variations in the write-off data as effectively as using the actual write-off values..

    c. Which version (IGe, IGv, or IGwo) should be used for a decision support tool, and explain why giving an example that helps explain your choice? (feel free to make a copy of one of the sheets and try changing the income data to demonstrate your answer, just write what you changed and the result rather than pasting the XLSX file into this document).

A. The choice between IGe, IGv, or IGwo for a decision support tool depends on the specific tool's objectives. If the tool's primary goal is to select the most informative features for making accurate predictions, IGe should be used. If the focus is on understanding the relationship between the income variable and the target variable, IGv(income) is appropriate. On the other hand, if the tool aims to assess the impact of write-offs on a customer's behavior or outcomes, IGwo is the right choice. And after changing the class of income variable(49975) to write-off from non-write off, there is no change in the over all result but the IGe(Information Gain with respect to the target variable) value decreased from 0.3033 to 0.2810.

    d. Is variance of Write-off value or its logical representation (1 & 0) even a valid choice for a decision support tree: why or why not.

A. The validity of using Write-off variance or a logical 1 and 0 representation in a decision support tree depends on the context and goals. Write-off variance is valid when it offers essential insights, but high variance can lead to overfitting. A binary representation is valid if it aligns with the decision process,

simplifying the model, but it might lose nuanced information. The choice should consider the specific use case, information needs, and trade-offs between model complexity and interpretability.

Sai Ganesh Pendela

2. Review the “table model” introduced in Chapter 5 (pg 114), answer the following questions:
    a. Explain what might happen when new data is tested against the model? Be sure to include discussion about the number of records in the table, the number of features, and feature types of the model.

A. In this scenario, the model performs very well on the training data, with almost zero errors, because it essentially memorizes it due to the large table (N rows). However, this is not the same, when new testing data is introduced, it consistently performs poorly because it's unable to generalize beyond the training set, indicating clear overfitting. And in this model, the number of features really matters. If you have too many features compared to the data points you're working with, it can make the problem of overfitting, where the model remembers things but doesn't really understand them. And the type of features you use is important too. If you go for complex or irrelevant ones, you're also making overfitting more likely. But if you stick to simple, relevant features, it helps the model do a better job of handling new data.

    b. Also, discuss how the “table model” should report its results and what is the danger in this approach? Use the Iris data set to help explain your answer.

A. In the scenario with the "table model," it's crucial to report both the training error and the testing (holdout) set error to provide a comprehensive assessment of the model's performance. Reporting solely the training error, which decreases as the model memorizes the training data, can be misleading and create a false impression of accuracy. The danger in this approach is that it may lead to overfitting, where the model struggles to generalize to new data. Even in the case of iris data set if you the same table model, it works good for the training data set but the model struggles to generalize to new data because the holdout error is constant i.e the table model is not actually learning from the data, it is actually memorizing the data.

    c. How does this “table model” compare functionally to a decision tree?

A. The "table model" and a decision tree both try to make predictions from data, but they work differently. The "table model" mainly memorizes data as it gets more complex, which can lead to problems with new data. On the other hand, a decision tree learns rules from the data to make predictions, making it better at handling new situations. If we give a new input which is not in the training set to the table model, it does not provide the exact output, whereas the decision tree gives exact output since the decision tree learns rules from the data. So, the "table model" is about memorizing, while a decision tree is about understanding and being able to handle different things. And the decision tree may not have as much as holding error as that of the table model.

Sai Ganesh Pendela

3. Discuss the differences between decision trees and logistic regression. Be sure to explain the basic methodology of both and how each model supports decision making.

A. Decision trees and logistic regression are both popular machine learning algorithms used for classification problems. Both algorithms have their own strengths and weaknesses, and the which algorithm to choose between them depends on the specific characteristics of the dataset and the problem at hand.

Decision Tree	Logistic Regression
Decision trees are a non-parametric supervised learning algorithm used for both classification and regression tasks	Logistic regression is a parametric statistical model used for binary and multi-class classification tasks.
Decision trees can handle both categorical and numerical features.	Logistic regression typically works with numerical features but can handle categorical data through encoding.
Decision trees use various criteria like Gini impurity, entropy, and information gain to determine the best attribute to split the data.	Logistic regression uses the logistic (sigmoid) function to transform a linear combination of input features into a probability score between 0 and 1.
Decision trees are prone to overfitting, especially when they become deep and complex.	It tends to be less prone to overfitting compared to decision trees, especially when the number of features is limited.
Decision trees select only one attribute individually in each step.	Logistic regression, on the other hand, uses all features at once.
Decision trees can handle missing values and outliers in the data much better than logistic regression.	Logistic regression, on the other hand, assumes that all features have non-zero values, so it can't handle sparse data.
Basic Methodology: Decision trees use a recursive, rule-based methodology to create a tree-like structure. They start with the entire dataset and split it into subsets based on significant attributes. Decision rules are formed at internal nodes, and you follow the path to reach a decision. This process continues until a stopping criterion is met. Decision trees are highly interpretable and suitable for complex decision boundaries. It uses Gini Index or entropy and IG to determine the best split attribute.	**Basic Methodology:** Logistic regression employs a sigmoid-based methodology to model the linear relationship between input features and the log-odds of a binary or multi-class event occurring. It transforms the linear output into a probability score, typically between 0 and 1, which represents the likelihood of the event. It assumes linearity between features and provides coefficients for each feature to interpret their impact
Support for decision making: Decision trees support decision making by providing clear rules for classification or regression. They allow users to follow a path in the tree to reach a decision based on the input features	**Support for decision making:** Logistic regression supports decision making by providing probabilities of class membership. A decision threshold can be set to classify instances into different classes based on these probabilities.

4. Answer the following questions regarding cross-validation:
    a. What is the reason to split the known data into training and test data?

A. The reason to split known data into training and test data is to assess the performance of a machine learning model while ensuring it can generalize well to new, unseen data. By separating the data, we can train the model on one the training set and evaluate its performance on the test set, which acts as unseen data. This process helps us detect and avoid issues like overfitting, where a model fits the training data too closely and fails to generalize. It also allows us to measure the model's ability to make accurate predictions on new, previously unseen data, which is the ultimate goal of machine learning model. Furthermore, the test set is essential for assessing various performance metrics. Cross-validation extends this idea by repeatedly splitting the data into training and test sets, providing a more robust assessment of a model's performance by averaging results over multiple splits.

    b. Why do we split the known data into multiple sets and how is this different from using a single split? Use the iris data set and model to help explain your answer.

A. We divide the known data into multiple subsets through cross-validation to thoroughly evaluate our machine learning model's performance. This differs from a single split because it offers a more robust assessment by testing the model on different parts of the data. For instance, take the Iris dataset with three different flower species. In a single split, the test set might accidentally have mostly one type of flower, leading to biased results. Cross-validation, like k-fold cross-validation, divides the data into multiple parts, enabling the model to be tested across various data segments. This approach ensures a better understanding of how well the model can perform under diverse conditions.

    c. And what could happen if the splits are not done randomly? Use the iris data set and model to help explain your answer.

A. If the splits in the data are not done randomly, there's a risk of introducing bias into the model evaluation. In the context of the Iris dataset, which contains three different flower species, non-random splits might lead to a situation where the same type of flower is predominantly present in the training or test set consistently, which might lead to perform poorly on other type of flowers. Random splits ensure that all data points have an equal chance of being in the training and test sets, offering a fair representation of the model's ability to generalize across various instances.

    d. How should one decide how many groups to use when splitting data?

A. When deciding how many groups to use for cross-validation data splitting, it's about finding a balance. More folds give a thorough assessment but need more computing power, while fewer folds are quicker but might provide more variable results. Your choice should consider your dataset's size, as smaller datasets can benefit from more folds. In simple terms, it's about figuring out the right balance to make sure we can accurately check how good a model is without making it too hard for the computer. To do this, we might need to try different things and think about the kind of data and problem we're working with.

    e. Can known data be used more than once and what would the expected result be?

A. In cross-validation, it's not a good idea to use the same data more than once. Doing so could make the model overly familiar with the data and give an unrealistically good impression of its abilities. It's

like training on the same data again and again and, answering those specific questions, but you won't know how well the model will perform on a new, different test data. Cross-validation ensures each piece of data is only used once, providing a fair and realistic assessment of how well the model can handle new, unseen data.

Sai Ganesh Pendela

5. Answer the following questions regarding a similarity approach to decision making:
    a. Describe the importance of the number of nearest neighbors used to make a determination in a k-NN algorithm.

A. The choice of the number of nearest neighbors (k) in a k-NN algorithm is crucial as it directly affects the model's accuracy, generalization capability, and sensitivity to noise. A smaller 'k' (e.g., k=1) makes the model overly sensitive to noise and individual data points. When 'k' is small, the algorithm essentially looks at the nearest neighbor, and this could lead to overfitting. Conversely, a larger 'k' (e.g., k=50) smooths the decision boundary and can lead to underfitting. Underfitting happens when the model is too simplistic to capture the complexities of the data. The decision boundary becomes overly smooth and might fail to capture important patterns in the data, resulting in poor performance, even on the training data. Finding the right 'k' value is about maintaining a balance between these extremes

    b. What happens when weighted scoring is used?

A. When weighted scoring is used in the k-NN algorithm, it means that instead of treating all neighbors equally, closer neighbors have a more significant influence on the prediction while more distant neighbors have less impact. Weighted scoring can enhance the model's performance, particularly when some neighbors are more relevant than others, ensuring that the k-NN algorithm gives more weight to the most informative data points, leading to more accurate predictions in scenarios where the distribution of data is uneven or there are varying degrees of influence among the neighbors.

    c. How is a k-NN decision similar to a decision tree?

A. k-NN and decision tree decision share a similarity in their rule-based nature. In k-NN, the decision is a straightforward rule: "Be assigned to the category that is the most prevalent among the k nearest neighbors". This approach mirrors the simplicity of a decision tree's logic, where decisions are made by following a series of if-else conditions based on attribute values. Both methods are transparent and provide clear, interpretable rules that explain how the model reaches its decisions. This common thing of rule-based decision-making makes both k-NN and decision trees make them similar. And both K-NN and decision tree are supervised machine learning algorithms used for both classification and regression tasks.

    d. How does a 1-NN classifier result in an overfit situation and how would a larger value of "k" change this?

A. A 1-NN classifier can overfit because it pays too much attention to small, noisy details in the training data, which might not represent the true underlying patterns. It essentially memorizes the training data and doesn't generalize well to new, unseen data. To avoid this, you can use a larger "k" (like 3 or 5), which means the model considers more neighbors when making predictions. This larger "k" smooths out the decision boundary and reduces sensitivity to individual data points, helping the model generalize better to new data. So, the larger "k" makes the model less prone to overfitting and more reliable in making predictions for different situations.

e. How would weighted-scoring for the 1-NN change this outcome?

A. Using weighted scoring for 1-NN would change the outcome by giving more importance to the nearest neighbor while considering its distance. Instead of just relying on the closest neighbor, weighted scoring would weigh it based on how close it is. This way, even if there's some noise or outlier in the closest neighbor, it wouldn't have an excessively strong impact, and the model would become a bit more robust, reducing the risk of overfitting. It essentially allows the 1-NN model to be less affected by anomalies and make better predictions.