K-Neighbors Classifier Assignment

Introduction

In this assignment, you will familiarize yourself with the foundational concepts of a simple yet powerful algorithm: the k-nearest neighbors classifier. You will be working with data related to online shopping behaviors, aiming to predict if a user will generate revenue (make a purchase) based on numerous attributes like the number of pages they visited, the duration of those visits, and more.

The provided template has skeletal functions with specific tasks outlined for you. Your challenge is to use your knowledge and the hints provided to fill in the gaps. More detailed instructions about the logic implemented in the assignment can be found at this link.

Instructions

1. **main function**
    - *train_test_split*: This function from scikit-learn will help you split your dataset into training and test subsets. Reflect on what data and parameters need to be passed here.
    - *Training the model*: Remember that you need a training dataset to train a model.
    - *Making Predictions*: Once your model is trained, think about which data you should use to make predictions.
    - *Evaluation*: Your evaluation will require comparison between the true labels and the predictions your model made.
2. **load_data function**
    - Each attribute in the dataset has a specified data type. It's crucial to ensure the data you load matches this type. For instance, consider what data type attributes like 'Administrative' or 'Informational_Duration' should have.
    - The provided *month_to_number* dictionary translates month abbreviations to numbers. Think about how and where this could be useful.
    - Pay attention to categorical data. How should you represent the 'VisitorType' or 'Weekend' attributes numerically?

```
$ python shopping.py shopping.csv
Correct: 4088
Incorrect: 844
True Positive Rate: 41.02%
True Negative Rate: 90.55%
```

Figure 1: Terminal output for purchase prediction.

3. **train_model function**

   - Remember, you're working with the k-nearest neighbors algorithm. Consider how you might specify the number of neighbors in the classifier and how you should use the training data.

4. **evaluate function**

   - Sensitivity and specificity are measures of your classifier's performance. Think about the definitions of these metrics and how you might compute them given the true labels and your model's predictions:
   - *sensitivity* should be a floating-point value from 0 to 1 representing the "true positive rate": the proportion of actual positive labels that were accurately identified.
   - *specificity* should be a floating-point value from 0 to 1 representing the "true negative rate": the proportion of actual negative labels that were accurately identified.

Conclusion

Run the code in your terminal with 'python shopping.py data.csv'. The data file can be explored with a spreadsheet application, like Excel, or a text editor.