515 HOMEWORK 4:
PREDICTING CROWDFUNDING SUCCESS

Your assignment is to create a tool that trains a decision tree machine learning model to perform the task of classifying crowdfunding posts. Your dataset of crowdfunding posts is taken from Kiva.com, which allows users to loan small amounts of money to small business ventures in developing countries. Some posts meet their funding goals and receive the loans, whereas others expire before doing so. The goal of your tool is to predict the success or failure of these posts as accurately as possible.

The dataset contains data on 6,300 posts. Your decision tree model will make predictions based on five factors:

- Length: the number of characters in the post.
- Number of pictures: the number pictures in the post.
- Loan amount: the amount of money requested by the post.
- Bonus credit eligibility: whether the post was eligible for any bonus or promotional opportunity on Kiva.com (yes/no).
- User favorite post: whether the post received many page views on Kiva.com (yes/no).

Your decision tree model will predict the loan status (funded/expired).

The dataset is available at https://dgoldberg.sdsu.edu/515/kiva_data_full.json. The data is formatted as a JSON array. An example of the formatting is below:

```
[
        {
                "length": 1036,
                "loan_status": "expired",
                "number_of_pictures": 2,
                "loan_amount": 700,
                "bonus_credit_eligibility": "yes",
                "user_favorite_post": "no"
        },

      ...

        {
                "length": 0,
                "loan_status": "funded",
                "number_of_pictures": 2,
                "loan_amount": 3150,
                "bonus_credit_eligibility": "yes",
                "user_favorite_post": "yes"
        }

                                                                                ]
```

Note that some of the variables described above are formatted as textual data (yes/no or funded/expired). You may need to transform these variables to a 1/0 format to train your decision tree model.

Your submission should consist of two parts:

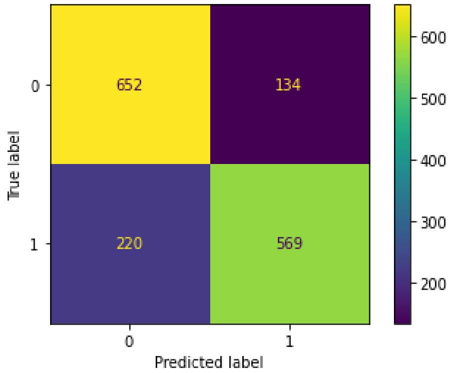
Part 1

In this program, you will train the decision tree machine learning model and evaluate its accuracy. Split your data into training and test data. No specific proportion of training versus test data is required. At the end of your program:

- Print out the accuracy score for your decision tree model. Your accuracy score may not exactly match the score below, as it depends on your exact training/test split.
- Show a confusion matrix for your decision tree model.
- Output a saved version of the decision tree model to a joblib file.
- You ***do not*** need to show the visual plot/tree for this assignment. The tree becomes quite large because there are many variables being considered. If you were to plot the decision tree, then it might be advisable to use `sklearn.tree.plot_tree(clf, max_depth = 3)` (the `max_depth` setting only shows the first few levels of the decision tree).

An example is below:

```
Accuracy: 0.7752380952380953
```



```
Decision tree model saved to kiva_decision_tree.joblib.
```

Part 2

In this program, ask the user for a series of inputs to load the decision tree model from a saved joblib file and predict the funding status for a new Kiva.com post. Then, load the decision tree model from the saved file and use it to make your prediction and print out the result. (Note: scikit-learn is expected a 2D list to make predictions, so you will need to reformat the user's inputs.) An example is below:

```
Enter the name of the decision tree file to load:
kiva_decision_tree.joblib

Enter the length of the post to predict: 1036
Enter the number of pictures in the post: 2
Enter the loan amount requested: 700
Enter the bonus credit eligibility (yes/no): yes
Enter whether the post was a user favorite post (yes/no): no

Based on the decision tree, the loan will not be funded.
```

Submit two separate Python files for this assignment: one for part 1 and one for part 2.

Some considerations as you write your programs:

- Consider the possibility that, when loading the dataset, some connection issue occurs (that is, a status code other than 200). Ensure that your code handles this case and provides the user with a helpful printout if it does occur.
- In part 2 of the assignment, handle any capitalization discrepancies in user inputs (e.g., "yes" versus "Yes" versus "YES").
- It is possible that your accuracy numbers may differ slightly from the values above due to differences in your training/test split; if your values are generally close, though, then this is not a problem.
- Ensure that your output is crisp, professional, and well-formatted. For example, ensure that you have used spaces appropriately and checked your spelling.
- Adding comments in your code is encouraged. At minimum, please use a comment at the start of your code to describe its basic functionality. In addition, for any functions or classes in your code, write appropriate docstrings. Ensure that your code would be as understandable as possible for a programmer working with your code for the first time.