Atharva Fulay - 1853414943
INF 552 Final Project

All of the files to create the models and visuals are provided in the zip file. The following python libraries are required:

- numpy
- pandas
- sklearn's RandomForestClassifier
- sklearn's DecisionTreeClassifier
- sklearn's shuffle
- sklearn's KFold
- sklearn's tree
- os
- matplotlib.pyplot
- subprocess' call

All you need to do is run `data_and_models.py`.

1. There are two files.
    a. `data_and_models.py`
        i. This file loads the data, executes the K-Fold cross validation, traditional training and testing (which generates the `dtc_results.txt` and `rfc_results.txt`), generates the optimal tree visual, confusion matrices, and the feature importances visual. It also calls analysis.py to create the other visuals used in the report.
    b. `analysis.py`
        i. This file will use the `dtc_results.txt` and `rfc_results.txt` to generate the analysis and visuals used in the report.
2. Folder structure:
    a. `data_and_models.py` and `analysis.py` should be in the current directory.
    b. `data_and_models.py` creates 3 folders:
        i. `data` – will contain the data (`data_and_models.py` will automatically move the `agaricus-lepiota.data` and `agaricus-lepiota.names` files if they are in the current directory)
        ii. `images` – all images will be placed in this folder
        iii. `res` – both `dtc_results.txt` and `rfc_results.txt` will be placed in this folder

If you choose to the the code, this is the format of the output:

```
---------- K-Fold Model Analysis ----------
Using K-Fold: average RFC accuracy (max_depth=4, max_features=sqrt) 0.9906461503038676
Using K-Fold: average DTC accuracy (max_depth=4, max_features=sqrt) 0.9487901344530686

---------- Generate Various models for visual analysis ----------
You can now look for "rfc_results.txt" and "dtc_results.txt" in the res folder. analysis.py
will make use of these.

---------- Generating Optimal Decision Tree Visual (see images folder) ----------
Check the images folder for "optimal_tree.png".

---------- Generating Optimal RFC Feature Importances Visual ----------
Check the images folder for "feature_importances_optimal_RFC.png".

---------- Confusion matrices for DTC and RFC ----------
RFC (max_features=sqrt, weighted):
Predicted     edible  poisonous
Actual
edible        4208    0
poisonous     0       3916

DTC (max_features=sqrt, weighted):
Predicted     edible  poisonous
Actual
edible        4208    0
poisonous     0       3916

Max-depth capped RFC on test data (max_features=sqrt, weighted, max_depth=5):
Predicted     edible  poisonous
Actual
edible        1026    71
poisonous     0       1027

Max-depth capped RFC on all data (max_features=sqrt, weighted, max_depth=5):
Predicted     edible  poisonous
Actual
edible        3940    268
poisonous     0       3916

Max-depth capped DTC on test data (max_features=sqrt, weighted, max_depth=5):
Predicted     edible  poisonous
Actual
edible        1002    95
poisonous     0       1027

Max-depth capped DTC on all data (max_features=sqrt, weighted, max_depth=5):
Predicted     edible  poisonous
Actual
edible        3840    368
poisonous     0       3916

---------- End data_and_models.py ----------
---------- Calling analysis.py ----------
---------- Generated all visuals (see images folder) ----------
---------- End analysis.py ----------
---------- End ----------
```