CS542 Final Challenge Assignment

In this final challenge assignment, you will be building your own machine learning solution to predict the price of an Airbnb rental given the dataset we have provided.

To submit your solution, you will upload your code (see below for submission instructions). You will make an **initial submission** with some model (a good one to start with can be Linear Regression). Due date for initial submission is **Nov 22, 5 pm EST**. You will get 10% of the score from a successful initial submission. The final deadline for this assignment is **Dec 2, 5 pm EST**.

Problem and dataset description

Pricing a rental property such as an apartment or house on Airbnb is a difficult challenge. A model that accurately predicts the price can potentially help renters and hosts on the platform make better decisions. In this assignment, your task is to train a model that takes features of a listing as input and predicts the price.

We have provided you with a dataset collected from the Airbnb website for New York, which has a total of 39,981 entries, each with 764 features. The data is split into a training set and a test set. You may use the provided data as you wish in development. We will train your submitted code on the same provided training set, and will evaluate it on yet another, hidden, test set.

We have already done some minimal data cleaning for you, such as converting text fields into categorical values and getting rid of the NaN values. To convert text fields into categorical values, we used different strategies depending on the field. For example, sentiment analysis was applied to convert user reviews to numerical values ('comments' column). We added different columns for state names, '1' indicating the location of the property. Column names are included in the data files and are mostly descriptive.

Also in this data cleaning step, the price value that we are trying to predict is calculated by taking the log of original price. Hence, minimum value for our output price is around 2.302 and maximum value is around 11.488.

All input features have already been scaled to [0, 1].

Folder structure:

Please download the data from the link posted on Piazza/Resources. Expected file structure for this project is shown in below:

```
.
├── Data
│   │
│   ├── data_cleaned_test_comments_X.csv
│   ├── data_cleaned_test_y.csv
│   ├── data_cleaned_val_comments_X.csv
│   └── data_cleaned_val_y.csv
│
├── Main
│   │
│   ├── model.py (initally) --> model_0000.py (while submission)
│   └── train_and_evaluate.py
│
```

```
└── README.md
```

Instructions to build your classifier:

1. Modify the `model.py` file name and postfix it with last four digit of your BU ID. For example, if your ID is U12345678, change your `model.py` file to `model_5678.py`.
2. Update the `ID_DICT` in `model.py` file with your info.
3. Modify `model.py` `Model` class to the model you will be building (do NOT change the class name). For the model, you will implement three functions: `preprocess`, `train`, and `predict`. The training and evaluation script `train_and_evaluate.py` will call these functions. **Please no not make any changes** to the `train_and_evaluate.py` file. Graders will be running these scripts automatically once you submit. The only thing you can change is the default BU ID in the argparser to make your debugging more convenient.
4. Test your model by running `python train_and_evaluate.py --bu_id XXXX` with your ID number.

An example linear regression model with random weights is provided to you in `model.py` file. Take a look at the file and replace the code with your own.

How is your code evaluated:

When we receive your submission, we will be running the `train_and_evaluate.py` code on your model, training on the same training set and testing against our hidden test set that is not released. Your score (we will use MSE as the primary metric) on the test set will be your performance measure. Your final grade will depend on the following criteria:

1. does the code run without errors?
2. does it follow rules (see below)?
3. is it original code (implemented by you)?
4. does it take a reasonable time to complete?
5. does it achieve a reasonable MSE?

Instructions to submit:

When you submit, you just need to upload your `model_XXXX.py` file (see Piazza for submission links).

Important:

- Use Python 3.8
- Only "import" the whitelisted libraries (see list below), your code MAY NOT use any other libraries
- Put your BUID in the model file e.g. `model_5678.py` and update the `ID_DICT` inside this file with your info
- Hardware constraints: we will evaluate on a CPU with 32GB and 16 cores
- Before uploading, please **test your code first with the provided evaluation script** to make sure it runs correctly.

- To get full points on the **initial** submission, the evaluation script must run under 10 min. without errors and must produce an MSE score (no specific value)

What is allowed and not allowed?

You are meant to implement your own machine learning model, so we are NOT allowing the use of machine learning libraries in your submitted code, such as (but not limited to):

```
pytorch
tensorflow
sklearn
keras
jax
...
```

You are ONLY allowed to import the following white-listed libraries:

```
numpy
sklearn.preprocessing  (but NOT sklean)
CVXOPT
pandas
...
```

You may NOT use additional datasets. This assignment is meant to challenge you to build a better model, not collect more training data, so please only use the data we provided.

GOOD LUCK!