How to add a model step-by-step:

1. Adding a model consists of 3 steps
    a. Ensure the train, test and validation csv data files exist.
    b. Ensure your model can integrate into the model pipeline framework
    c. Ensure you run your model using the model pipeline framework
2. Ensuring the train, test and validation csv data files exist.
    a. Check the repo for your like train_1K.CSV or test_10K.csv, etc
    b. If your data file doesn't exist, add an entry to gen_data_info.csv config file, then run Jupyter notebook test_data_generator.ipynb. Baring any error, this should generate your data file you specified in gen_data_info.csv. Ensure you have an entry for train, test and validation.
3. Ensure your model can integrate into the model pipeline framework
    a. Your model primarily needs to inherit from the SASentimentModel abstract class
    b. And then implement all the abstract methods which your model needs anyway.
4. Ensure you run your model using the model pipeline framework
    a. Once you have your model developed and ready to test, then simply add an entry to the model_config.csv
    b. The first column is the name of your model. You can name it Supercalifragilisticexpialidocious. It is only used for display purposes
    c. The 2nd column is the name of the module. This is the file name of your model. For example, you can name your file sa_self_attention.py. Then the 2nd column should be sa_self_attention
    d. The 3rd column is the class name. This is the name of your model class. For example, "class SASelfAttentionModel:" then the 3rd column should be SASelfAttentionModel\
    e. The 4th column is any model params you like to use in your model. For example, vocab_size, max_sequence_length, epoch, etc. Then the 4th column should be a quoted string like "vocab_size=10000,max_sequence_length=100,epoch=10"
5. After you have configured the model_config.csv file, then
    a. Run either run Jupyter notebook
        i. test_model_pipeline.ipynb that runs ALL the models defined in the model_config.csv file or
        ii. test_single_model_pipeline.ipynb and modify the 2 parameters, model_module_name and model_class_name.
        iii.
```
sa_model_pipeline = SAModelPipeline()

model_module_name = "SASelfAttentionModel"
model_class_name = "SASelfAttentionModel"
sa_model_pipeline.run_single_model_pipeline(model_module_name,
                                            model_class_name)
```
        iv. This will run your model instance calling all the abstract methods you defined in your class.
6. That's it.