README.md

Steps to run the experiments in section 5, subsections RQ2 and RQ3:

Step 1. Install lale. In the supplemental material, we have provided a zip with lale code. Unzip lale.zip and execute the commands below:

```
        cd lale
        pip install .[full]
```

For a quick check, do the following in the `lale` directory:

```
        export PYTHONPATH=`pwd`
        python -m unittest test.test_core_classifiers.TestLogisticRegression
```

The output should look like::

```
        Ran 12 tests in 0.931s
        OK
```

Step 2. RQ2: The `RQ2` directory contains the Python scripts as well as results for the experiments reported in subsection RQ2. For each of the columns in Table 2, there is a Python script for a single run of the experiment. The script can be run as:

```
        python <name_of_the_script>
```

Here is the table mapping the experiment name (column name) in Table 2 to its script:

Experiment Name	Script Name
AUTOSKL	auto_sklearn.py
LALE-AUTO	lale_pipeline.py
LALE-TPOT	tpot_grammars.py
LALE-AD3M	alphad3m_grammar.py
LALE-ADB	lale_pipeline_adaboost.py
ASKL-ADB	compare_auto_sklearn_adaboost.py

The results are saved in the logs directory. The `setup.py` in the directory has the settings such as time budget for optimization, datasets used etc.

Step 3. RQ3: The `RQ3` directory contains the Python scripts as well as results for the experiments reported in subsection RQ3. For each of the settings of Table 3, there is a bash script that runs the appropriate Python script on the Phoneme dataset 5 times. You can run the bash script as follows:

```
        ./<bash_script_name>
```

The results are saved in the logs directory and `analyze_small_pipeline_results.py` can take the different experiment names as input and summarize the results.

Note that the implementation of optimizer backends ADMM and Hyperband used in this experiment are not in open source, so it won't be possible to run these settings.