Thermopred documentation

This repository contains the official data, algorithms, and ML models present in the paper "`AI-Enhanced Quantum Chemistry Dataset for Thermochemical Properties of API-Like Compounds and Their Degradants`". The current structure of the repository is as shown in the image below.

The `dataset` folder contains the complete database used in the article, where the data on thermochemical and quantum properties are arranged, as well as the optimized string structure in XYZ format. The `docs` folder contains documentation files and instructions on how to use the data/algorithms. The `Thermopred` folder contains models trained to predict Gibbs free energy and enthalpy, as well as algorithms to run these models and make predictions. The other files in the root of the directory are configurations needed to install this algorithm as a python package.

```
.
├── dataset
│   └── FullDataset.csv
├── docs
│   ├── Documentation.md
│   └── Documentation.pdf
├── LICENSE
├── MANIFEST.in
├── README.md
├── requirements.txt
├── setup.cfg
├── setup.py
└── Thermopred
    ├── Enthalpie.py
    ├── GibbsEnergy.py
    ├── __init__.py
    ├── models
    │   ├── MLP_gibbs.sav.tar.gz
    │   ├── RF_gibbs.sav.tar.gz
    │   └── XGB_gibbs.sav.tar.gz
    └── models_enthalpie
        ├── MLP_gibbs_enthalpie.sav.tar.gz
        ├── RF_gibbs_Enthalpie.sav.tar.gz
        └── XGB_gibbs_Enthalpie.sav.tar.gz
```

How to use

Download the repository manually or via git:

```
$ git clone https://github.com/jeffrichardchemistry/thermopred
```

Enter the `thermopred` directory and run the following command to install the python package:

```
$ cd thermopred

$ python3 setup.py install
```

Once this is done, it is now possible to use the package by simply importing the modules. Import the modules as described below and pass a smiles for prediction.

```python
from Thermopred.Enthalpie import EnthalpieEnergy
from Thermopred.GibbsEnergy import GibbsFreeEnergy

smiles='CN1C=CN(CCCN(c2cc(Cl)ccc2O)c2ccccc2S)CC1'

ee = EnthalpieEnergy()
result_enthalpie = ee.predict(smiles)

gfe = GibbsFreeEnergy()
gfe.predict(smiles=smiles)
```