CHEM452/CHBE413

Chemical Data Science and Engineering

Homework 2

Due Date: September 11th, 2025

Please complete and submit this homework as a Jupyter notebook. *For every plot your axes must be labeled, and if there are multiple data sets on the same plot, a legend must be provided!*

1. **Processing and Visualizing Molecular Datasets with Pandas. (**dataset:*Tm\_200\_subset\_raw.csv)****.*** “*Tm\_200\_subset\_raw.csv*” contains the melting points (Tm) of 200 drug-like molecules, as well as the chemical structure (encoded as a SMILES string – more on that later), total number of atoms, dipole moment, and quadrupole moment for each molecule. In this problem we will work on processing and analyzing the raw Tm dataset for which errors, duplicates, or gaps may exist.

Before getting started, make sure you load the necessary Python libraries. In this homework we will make use of Pandas and RDKit - run the following commands at the top of your jupyter notebook in Colab to make sure they are installed. It should take <1 minute to install.

!pip install rdkit-pypi

import rdkit

from rdkit import Chem

from rdkit.Chem import AllChem

from rdkit.Chem import Draw

import pandas as pd

1. **(3 points)** Using pandas (you can reference it as “pd” based on how it was imported), load in the “*Tm\_200\_subset\_raw.csv*” as a data frame using *df =pd.read\_csv(“Tm\_200\_subset\_raw.csv”)*. Look at the header of the data frame you have imported using *df.head()* to make sure it loaded properly. First, let’s make sure there are no duplicates in the dataset (e.g. make sure the data for one molecule was not listed more than once – you would be surprised at how common this error is!). How many duplicates exist in the dataset? Pandas can automatically search for, and delete, duplicates - drop all identified duplicates from your dataset using the *df.drop\_duplicates()* command.
2. (**3 points)** Using the *df.info()* method, check to see if all of the remaining rows and columns you have uploaded are populated. Often datasets will be missing property values because a measurement could not be made. You can identify all columns with missing entries using *df.isnull()*, and you can count how many values are missing for each column using *df.isnull().sum()*. How many entries are missing from each column in the dataset?
3. (**3 points)** One can deal with these missing data points in a few different ways. One option is to fill the missing entries with imputed values from the given column (using the mean value of the associated column is common). What are the mean values of each column in the data set using *df[‘columnname’].mean()?* Why or why not might filling in the missing entries with the mean values might be a reasonable approximation? More info on imputation can be found here: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779>.
4. **(3 points)** As you might have suspected from part (c), in addition to missing values in the data, there were a few input errors in each column (some data was repeated, some was not included, and some was simply improperly reported). Plot each column as a histogram to see the distribution of your data (“*df[‘columnname’].plot(kind=’hist’, bins=20”)*. Next compute the statistics of your continuous variable columns using *df.describe()*. Use the standard deviations of each column (and common sense!) to justify whether you believe a molecule to be improperly labeled, and then drop those molecules (rows) from the dataframe. Clearly state the SMILES strings of the dropped molecules.
5. **(3 points)** The data set has three numerical features (number of atoms, dipole, quadrupole) with which to try to predict Tm. Let’s first understand how correlated our different columns (features) are. We can accomplish this by computing the correlations between all features via a correlation matrix amongst all of the columns using *df.corr(method=‘pearson’)*. What two columns are the most correlated? The least? Are any anti-correlated?
6. **(2 points)** Visual inspection of data is always a good idea! Plot Tm against each molecule feature (column) independently. Which feature exhibits the strongest visual correlation with the Tm? Is this consistent with the computed correlation coefficients in (e)?
7. (**3 points)** Select the SMILES strings for the three molecules in your dataset with the lowest Tm, and the three molecules in your dataset with the highest Tm. Using their SMILES strings, plot the 2D depiction of each molecule in your Jupyter notebook using the following commands. Here is an example for a molecule not contained in your dataset.

smiles = 'COC(=O)c1c[nH]c2cc(OC(C)C)c(OC(C)C)cc2c1=O'

mol = Chem.MolFromSmiles(smiles)

Draw.MolToImage(mol)

Identify (at least) two qualitative molecular descriptors (here are some ideas: types of bonds, types of atoms, types of rings, certain types of atoms, etc) that appear to correlate with increasing Tm, and justify (chemically) why these choices might correlate with Tm.