AI and Drug Discovery

Assignment # 3

Exploratory Data Analysis and PubChem Fingerprint Calculation

Submitted By: Syeda Rabia Gillani

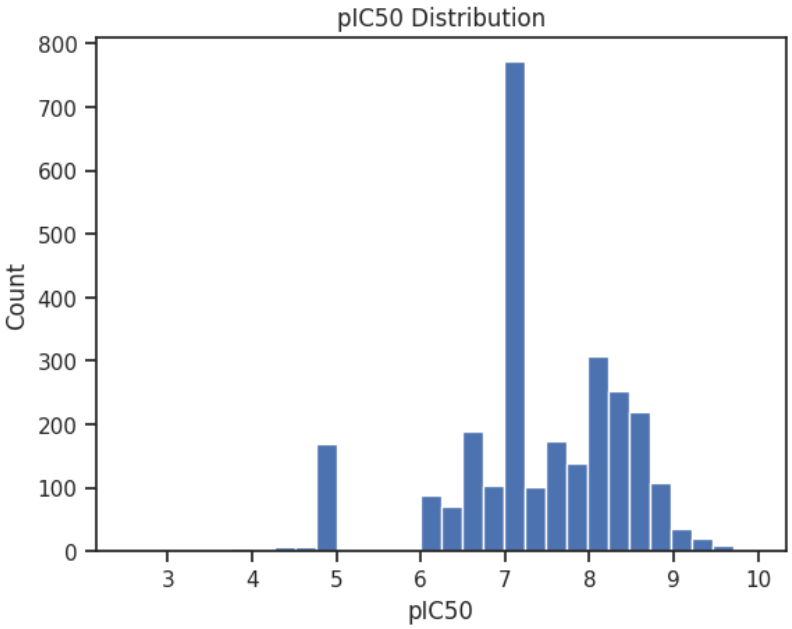
Summary

For Exploratory Data Analysis (EDA) of processed bioactivity dataset of FGFR2, I imported the required libraries, uploaded and loaded the ChEMBL bioactivity CSV, cleaned the dataset by removing missing values, converting IC50 (standard_value) to numeric, and dropping "intermediate" entries. Then I handled duplicates by grouping on canonical_smiles and taking the median IC50 per molecule. After that, I converted IC50 (nM) into pIC50 using $-\log_{10}(IC50\times10^{-9})$, relabeled compounds as **active** ($pIC50 \geq 6$) or **inactive** ($pIC50 < 6$), and confirmed no duplicate SMILES remained. I visualized the pIC50 distribution and the class counts, installed RDKit to calculate Lipinski descriptors (MW, LogP, H-bond donors, H-bond acceptors), merged these descriptors back into the dataset, and explored chemical space using barplots, boxplots, and an MW vs LogP scatter plot (colored by class and sized by pIC50). Finally, I ran Mann–Whitney U tests to check whether descriptor distributions differ between active and inactive compounds, saved the statistical outputs to CSV, exported the plots as PDFs, and bundled the results for download.

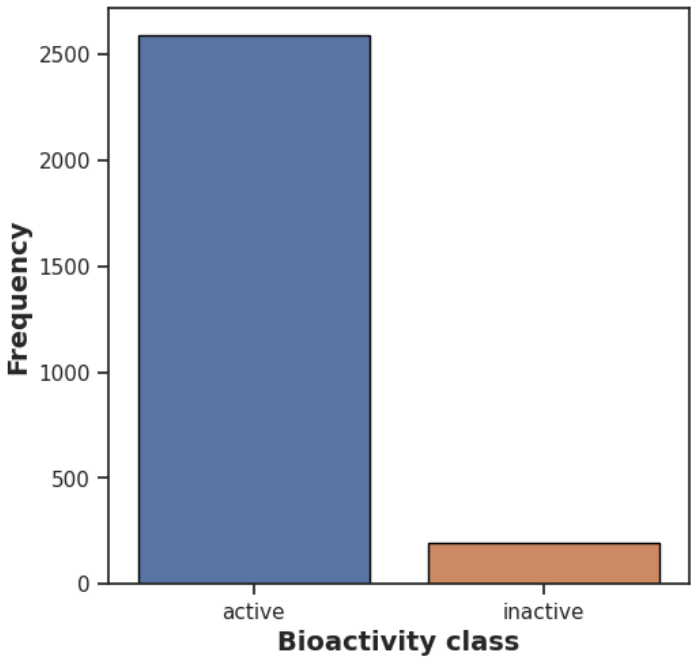
For PubChem fingerprint calculation, I installed **PaDELpy**, loaded my cleaned dataset (df_lipinski.csv), and extracted the **canonical SMILES** (and IDs) to create a **.smi** file (smiles_chembl.smi). Then I used padeldescriptor() to generate **PubChem fingerprints** for each molecule (a large set of binary fingerprint columns saved into pubchem_fingerprints.csv). I read that fingerprint file back into pandas, and **combined** it with my key labels/targets (molecule_chembl_id, bioactivity_class, pIC50) by resetting indices to keep rows aligned. Finally, I saved the merged dataset as a **QSAR-ready ML file** (QSAR_dataset.csv) for model training. I also downloaded PaDEL XML configs from GitHub to compute **other fingerprint types** (by selecting the required XML file, e.g., Substructure fingerprints, and running padeldescriptor() again).

Key EDA findings + visualizations:

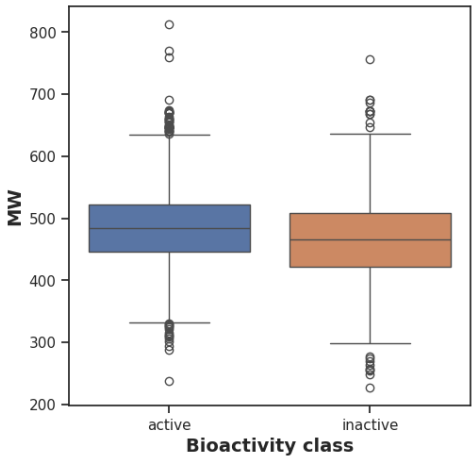
I first looked at how the activity values are spread out. The pIC50 histogram shows most compounds falling around the mid–high range (roughly ~7–8).

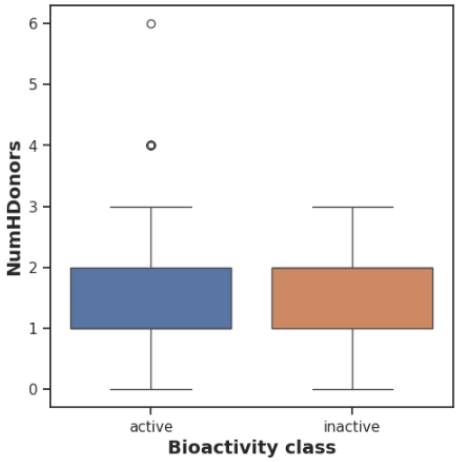


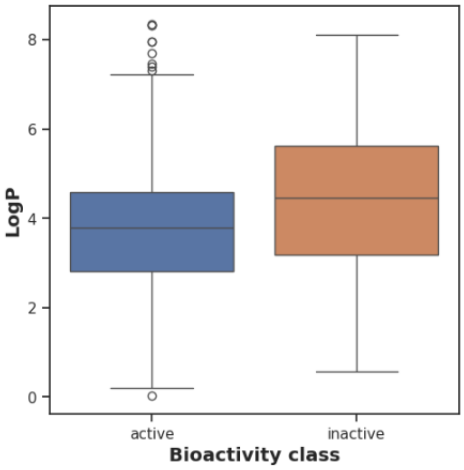
The class count plot also makes it clear the data is imbalanced, there are a lot more active compounds than inactive ones.

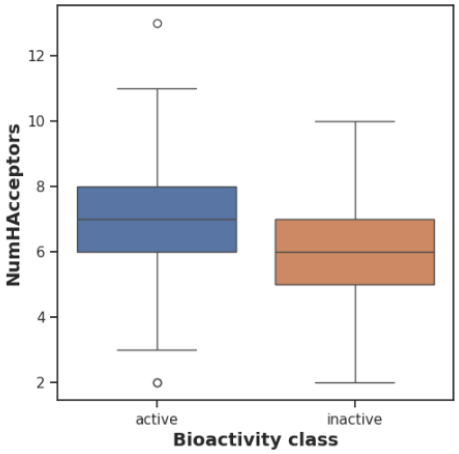


Then, using boxplots and the MW vs LogP scatter plot, I could see that actives and inactives don't sit in exactly the same chemical space, which hints that these properties might actually relate to activity.









Lipinski + 2D descriptor stats:

After calculating Lipinski descriptors (MW, LogP, HBD, HBA), I tested whether these features differ between active and inactive groups using the **Mann–Whitney U test**. All four came out **statistically significant** at $\alpha = 0.05$ (MW, LogP, HBD, and HBA all had p-values < 0.05). For 2D descriptors, the **PubChem fingerprints** gave me **881 binary fingerprint bits/features** per molecule.

Fingerprint method + why I used it:

I calculated fingerprints using **PaDEL-Descriptor through the PaDELpy Python wrapper**, generating **PubChem fingerprints** directly from SMILES. I picked this approach because it's a very common and accepted method in QSAR work, it's reproducible, and it produces a clean fixed-length feature vector that's easy to feed into machine learning models.