QSAR Modeling of Bioactivity using RDKit Descriptors

✅ Phase 0: **Preparation**

📦 Tools & Setup

- **Python environment**: conda or venv
  Install packages:

  ```
  pip install rdkit scikit-learn pandas matplotlib seaborn xgboost shap
  ```

- **IDE**: Jupyter Notebook or VSCode with Jupyter extension
- **Dataset**: Get activity data (SMILES + IC_{50}/Ki) from ChEMBL or PubChem.

📁 Project Folder Structure (for GitHub)

```
QSAR-Bioactivity-Prediction/
│
├── data/
│   ├── raw/            # Original SDF or CSV from ChEMBL
│   └── processed/      # Cleaned, ready-to-use CSV
│
├── notebooks/          # Jupyter notebooks
│   ├── 01_data_prep.ipynb
│   ├── 02_descriptor_calc.ipynb
│   ├── 03_modeling.ipynb
│   └── 04_evaluation.ipynb
│
├── src/                # Python functions for reuse
│   └── utils.py
│
├── results/            # Plots, feature importance, etc.
│
├── README.md           # Project description and instructions
├── requirements.txt    # For reproducibility
└── .gitignore          # Ignore data checkpoints, etc.
```

🧱 Phase 1: **Data Preparation**

Goal: Load raw bioactivity data → clean it → prepare SMILES and target_value columns.

Tasks:

- Download bioactivity data for a protein target (e.g., HIV RT, JAK2, hERG).
- Keep only valid SMILES, remove stereoisomers or duplicates if needed.
- Convert IC_{50} or Ki to **pIC_{50}**:

  pIC50=−log10(IC50 in mol1 M)\text{pIC}_{50} = -\log_{10}\left(\frac{\text{IC}_{50} \text{ in mol}}{1\,\text{M}}\right)pIC50=−log10(1MIC50 in mol)

✅ Save cleaned file as processed/bioactivity_data.csv

🧪 Phase 2: **Descriptor Calculation with RDKit**

Goal: Convert SMILES → numerical descriptors.

Options:

- Use rdkit.Chem.Descriptors for physicochemical descriptors
- Use Morgan Fingerprints:

  AllChem.GetMorganFingerprintAsBitVect(mol, radius=2, nBits=1024)

✅ Create a dataframe: X = descriptors, y = pIC50

🤖 Phase 3: **Model Training**

Goal: Train and compare regression models

Models:

- Linear Regression (baseline)
- Random Forest Regressor
- XGBoost Regressor

Validation:

- Split data: train_test_split
- Use cross_val_score (with R^2, MAE, RMSE)
- Plot:
  - Actual vs Predicted
  - Residuals

    - Feature importance (for RF/XGB)

✅ Save best model (e.g., using joblib)
✅ Save plots to /results/

📈 Phase 4: **Evaluation**

- **Metrics**:
    - R² (fit quality)
    - MAE, RMSE (error magnitude)
- **Plots**:
    - y_pred vs y_true
    - residuals
    - SHAP or permutation feature importance

✅ Write a Markdown cell summary inside the notebook.

🚀 Phase 5: **Packaging for GitHub**

Tasks:

- Write a clean README.md:
    - Project goal
    - Dataset used (ChEMBL, target ID)
    - Model pipeline
    - Key results
    - Example plots
- Create requirements.txt:

  ```
  pip freeze > requirements.txt
  ```

- Add .gitignore:

  ```
  __pycache__/
  .ipynb_checkpoints/
  data/raw/
  *.pyc
  ```

- Push to GitHub:

  ```
  git init
  git remote add origin https://github.com/yourusername/QSAR-Bioactivity-Prediction.git
  git add .
  git commit -m "Initial commit"
  git push -u origin master
  ```

📘 Optional Enhancements

- Add **consensus modeling** (like ISIDA_QSPR)
- Use **applicability domain analysis**
- Try **classification version** (active vs inactive threshold)
- Extend to **multitask QSAR** or **transfer learning**