**1. Model Selection and Training**

- The application provides a user interface to select different types of machine learning problems: **Classification**, **Regression**, or **Clustering**.
- After the user uploads a dataset and selects the target variable, they can choose which machine learning models to train from a predefined list.

**2. Model List**

- **Classification models**: Logistic/Linear Regression, Random Forest, XGBoost, CatBoost, Support Vector Machine (SVM), K-Nearest Neighbors (KNN).
- **Regression models**: Logistic/Linear Regression, Random Forest, XGBoost, CatBoost, Support Vector Machine (SVR), K-Nearest Neighbors (KNN).
- **Clustering algorithms**: K-Means, Hierarchical Clustering.

**3. Model Parameters**

- For each model, the user can configure hyperparameters. These include:
    - **Logistic/Linear Regression**: max_iter (maximum iterations).
    - **Random Forest**: n_estimators (number of trees).
    - **XGBoost**: n_estimators (number of boosting rounds).
    - **CatBoost**: n_estimators (number of boosting rounds).
    - **SVM**: kernel (type of kernel), C (regularization parameter).
    - **KNN**: n_neighbors (number of neighbors).
    - **K-Means** and **Hierarchical Clustering**: n_clusters (number of clusters) and, for hierarchical, linkage (type of distance linkage).

**4. Training the Models**

- After the user selects the models and their parameters, the code trains the chosen models using the provided dataset.
- **Classification/Regression** models are trained with train_test_split, splitting the dataset into training and testing sets.
- **Clustering** models are directly applied to the entire dataset without splitting since clustering does not require labeled data.

For example:

- **Logistic/Linear Regression**: LogisticRegression() or LinearRegression() from sklearn are used for classification or regression tasks, respectively.
- **Random Forest**: RandomForestClassifier or RandomForestRegressor is used depending on the problem type.
- **XGBoost**: XGBClassifier or XGBRegressor is used for classification or regression tasks.

- **CatBoost**: CatBoostClassifier or CatBoostRegressor for classification or regression tasks.
- **SVM**: SVC or SVR from sklearn is used for classification or regression tasks.
- **KNN**: KNeighborsClassifier or KNeighborsRegressor is used for classification or regression tasks.

**5. Model Evaluation**

- After the models are trained, their performance is evaluated using appropriate metrics:
    - **For Classification**: Metrics like accuracy, precision, recall, F1-score, and confusion matrix are computed.
    - **For Regression**: Metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and $R^2$ score are used.
    - **For Clustering**: The **Silhouette Score** is calculated to evaluate the quality of clustering.
- The evaluation results are displayed as tables and bar plots, allowing users to compare the performance of different models.

**6. Model Download**

- Once the models are trained and evaluated, users can download the trained models as .pkl files using the pickle library. This allows them to store the models and use them for future predictions.

**7. Implementation of Specific Models**

- **Logistic/Linear Regression**: Uses the LogisticRegression and LinearRegression classes from sklearn for classification and regression, respectively.
- **Random Forest**: Uses the RandomForestClassifier and RandomForestRegressor classes to create random forests for classification and regression.
- **XGBoost**: The XGBClassifier and XGBRegressor from the xgboost library are used for classification and regression, respectively.
- **CatBoost**: The CatBoostClassifier and CatBoostRegressor are used for classification and regression, respectively, from the catboost library.
- **SVM**: The SVC (Support Vector Classification) and SVR (Support Vector Regression) from sklearn.svm are used.
- **KNN**: The KNeighborsClassifier and KNeighborsRegressor from sklearn.neighbors are used for classification and regression.
- **Clustering**: KMeans from sklearn.cluster and AgglomerativeClustering are used for clustering tasks.

**8. Model Pickling**

- Trained models are pickled (serialized) using pickle.dumps() and saved as .pkl files for download. This allows users to store the models and use them later without retraining.