**1. Data Preprocessing and Balancing**

* **PCA (Principal Component Analysis)**:
  + Reduces the dimensionality of data while preserving most of the variance.
  + Useful for feature reduction to improve model efficiency.
* **Sampling Techniques** (to handle imbalanced datasets):
  + **RandomOverSampler**: Balances the dataset by oversampling minority classes.
  + **RandomUnderSampler**: Balances the dataset by undersampling majority classes.
  + **SMOTEENN**: A combination of oversampling using SMOTE and cleaning using Edited Nearest Neighbors.
* **OrdinalEncoder**:
  + Encodes categorical features into numerical values.

**2. Models for Classification and Regression**

The application supports several algorithms for both **classification** and **regression** tasks:

* **Logistic Regression / Linear Regression**:
  + Logistic Regression is used for binary/multiclass classification.
  + Linear Regression is used for predicting continuous values.
* **RandomForestClassifier / RandomForestRegressor**:
  + Ensemble methods using multiple decision trees to improve accuracy.
  + Classification: Used for predicting categorical outcomes.
  + Regression: Used for continuous value prediction.
* **XGBClassifier / XGBRegressor (XGBoost)**:
  + Gradient boosting algorithm known for high performance.
  + Efficient for both classification and regression tasks.
* **CatBoostClassifier / CatBoostRegressor**:
  + Boosting algorithm optimized for categorical features.
  + Provides state-of-the-art performance with minimal hyperparameter tuning.
* **SVC / SVR (Support Vector Machine)**:
  + SVC: Used for classification tasks, supports different kernels (linear, RBF, etc.).
  + SVR: Used for regression tasks, helps find the best-fit hyperplane.
* **KNeighborsClassifier / KNeighborsRegressor (KNN)**:
  + Classification and regression using the k-nearest neighbors algorithm.
  + Non-parametric method based on feature similarity.

**3. Clustering Algorithms**

For unsupervised learning, the application provides:

* **KMeans**:
  + Partitions data into clusters based on feature similarity.
* **AgglomerativeClustering**:
  + Hierarchical clustering that merges data points based on linkage criteria (e.g., ward, complete, average).

**4. Evaluation Metrics**

The application uses the following metrics for model evaluation:

* **Classification**:
  + **Accuracy Score**: Percentage of correct predictions.
  + **Precision, Recall, F1 Score**: Metrics to evaluate the quality of predictions, especially for imbalanced datasets.
  + **Confusion Matrix**: Shows true positives, true negatives, false positives, and false negatives.
* **Regression**:
  + **Mean Absolute Error (MAE)**: Measures the average absolute error between predicted and actual values.
  + **Root Mean Squared Error (RMSE)**: Indicates the square root of the average squared differences.
  + **R² Score**: Indicates the proportion of variance explained by the model.
* **Clustering**:
  + **Silhouette Score**: Measures how well samples are clustered, with higher values indicating better-defined clusters.

**5. Model Downloading**

* Trained models can be downloaded using the **pickle** library, allowing users to save models locally for future use.

**Overall Flow**

1. Users upload a CSV dataset.
2. The application performs data visualization and summarization.
3. Users can clean the data, balance the dataset, and apply dimensionality reduction.
4. Multiple models can be trained simultaneously based on user-selected parameters.
5. Users can evaluate the models and download the trained models as .pkl files.

This AutoML web application simplifies the end-to-end process of machine learning, making it accessible to non-experts by automating common ML workflows.