In [1]:
```
import pandas as pd
import numpy as np
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.model_selection import KFold
```

In [2]:
```
# Load the dataset
data = pd.read_csv('1M_ahead_dataset.csv')
```

In [3]:
```
# Separate predictors and target
X = data.drop(['Yt.1M'], axis=1)
y = data['Yt.1M']
```

In [4]:
```
# Set up 5-Fold cross-validation
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

In [5]:
```
fold_metrics = []
fold_counter = 1
```

In [6]:
```
# Loop over each fold for cross-validation
for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Initialize the XGBoost regressor.
    # n_estimators, learning_rate, and max_depth can be tuned further.
    model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=3,
                         random_state=42, objective='reg:squarederror')

    # Fit the model on training data
    model.fit(X_train, y_train)

    # Make predictions on the test data
    y_pred = model.predict(X_test)

    # Compute evaluation metrics
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mse)

    fold_metrics.append({
        'Fold': fold_counter,
        'MSE': mse,
        'RMSE': rmse,
        'MAE': mae
    })

    # Output feature importances for the current fold
    print(f"\nFold {fold_counter} Feature Importances:")
    print(model.feature_importances_)

    # Output performance metrics for the current fold
    print(f"Fold {fold_counter} -- MSE: {mse:.4f}, RMSE: {rmse:.4f}, MAE: {mae:.4f}")

    fold_counter += 1
```

```
Fold 1 Feature Importances:
[0.07284126 0.04739914 0.12152141 0.12797    0.11297792 0.04593138
 0.07605208 0.06437192 0.0735223  0.05396804 0.04417563 0.10472929
 0.05453971]
Fold 1 -- MSE: 0.0148, RMSE: 0.1217, MAE: 0.0762

Fold 2 Feature Importances:
[0.07091007 0.06809639 0.06560018 0.10606415 0.08495652 0.11408972
 0.06963606 0.05765824 0.06656368 0.04732427 0.05483212 0.0910652
 0.10320339]
Fold 2 -- MSE: 0.0179, RMSE: 0.1339, MAE: 0.0769

Fold 3 Feature Importances:
[0.06444621 0.08630675 0.069647   0.11406364 0.08389287 0.04214677
 0.09021536 0.08517633 0.08107425 0.05019517 0.05158941 0.08301485
 0.09823138]
Fold 3 -- MSE: 0.0142, RMSE: 0.1190, MAE: 0.0717

Fold 4 Feature Importances:
[0.06613237 0.06292125 0.05082086 0.09558128 0.11655889 0.04210566
 0.09157661 0.08149672 0.10137121 0.04027284 0.08843628 0.08328968
 0.07943631]
Fold 4 -- MSE: 0.0126, RMSE: 0.1123, MAE: 0.0767

Fold 5 Feature Importances:
[0.08219527 0.08289951 0.04258306 0.12431171 0.09923558 0.05599408
 0.06913089 0.05772585 0.1318689  0.05472418 0.06035369 0.08384348
 0.05513385]
Fold 5 -- MSE: 0.0193, RMSE: 0.1388, MAE: 0.0763
```

In [7]:
```python
# Summarize the results in a DataFrame
results_df = pd.DataFrame(fold_metrics)
print("\nOverall Cross-Validation Results:")
print(results_df)
```

```
Overall Cross-Validation Results:
   Fold       MSE      RMSE       MAE
0     1  0.014820  0.121738  0.076201
1     2  0.017918  0.133857  0.076943
2     3  0.014157  0.118981  0.071657
3     4  0.012607  0.112283  0.076724
4     5  0.019259  0.138777  0.076282
```