In [1]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
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]:

```
# Assume that 'Yt.1M' is the target and the rest are predictors
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]:

```
# Initialize list to store metrics for each fold
fold_metrics = []
```

In [6]:

```
# Loop over each fold
fold = 1
for train_index, test_index in kf.split(X):
    # Split the data into training and testing sets for this fold
    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 and train the linear regression model
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Print coefficients and intercept for this fold
    print(f"\nFold {fold} Model Coefficients:")
    print("Coefficients:", model.coef_)
    print("Intercept:", model.intercept_)

    # Make predictions on the testing set
    y_pred = model.predict(X_test)

    # Compute MSE, RMSE, and MAE
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mse)

    # Store metrics for this fold
    fold_metrics.append({
        'Fold': fold,
        'MSE': mse,
        'RMSE': rmse,
        'MAE': mae
    })

    # Print metrics for the current fold
    print(f"Fold {fold} -- MSE: {mse:.4f}, RMSE: {rmse:.4f}, MAE: {mae:.4f}")
    fold += 1
```

```
Fold 1 Model Coefficients:
Coefficients: [ 8.57399926e-02 -1.34475893e-02  1.85860291e-02  5.56688426e-02
 -2.43902121e-15  5.91034090e-03 -5.49367857e-08 -7.69700130e-04
 -8.47339872e-03  6.85459054e-02 -3.18775253e-04  1.73496342e-03
 -9.46106144e-04]
Intercept: 0.004637620298440101
Fold 1 -- MSE: 0.0157, RMSE: 0.1253, MAE: 0.0714

Fold 2 Model Coefficients:
Coefficients: [ 8.84757948e-02 -2.53638299e-02  3.47261896e-02  6.41062698e-02
 -4.14529522e-14  4.00549021e-03 -6.09405973e-08 -5.07895493e-04
 -7.01499591e-03  7.23376854e-02 -8.22124606e-05  2.19871161e-03
 -4.93436309e-03]
Intercept: -0.004677451499930985
Fold 2 -- MSE: 0.0219, RMSE: 0.1479, MAE: 0.0785

Fold 3 Model Coefficients:
Coefficients: [ 5.43835439e-02 -1.31762005e-02  2.09082195e-02  6.06720155e-02
 -2.09034179e-15  4.03700707e-03 -8.14764415e-08 -6.08796137e-04
 -1.10624430e-02  7.91777687e-02 -1.15049165e-04  1.62657381e-03
  2.83635510e-02]
Intercept: -0.0006663362703808878
Fold 3 -- MSE: 0.0125, RMSE: 0.1120, MAE: 0.0699

Fold 4 Model Coefficients:
Coefficients: [ 5.78539920e-02 -1.13816159e-02  1.83593967e-02  6.07927545e-02
 -2.85362012e-15  6.45619753e-03 -9.41503572e-08  7.11465118e-05
 -1.18115251e-02  5.72995141e-02 -3.46239371e-04  1.45269143e-03
  1.75155611e-02]
Intercept: 0.01266822605462483
Fold 4 -- MSE: 0.0115, RMSE: 0.1074, MAE: 0.0726

Fold 5 Model Coefficients:
Coefficients: [ 6.15726446e-02  1.92123301e-03  1.18041984e-03  6.21004993e-02
 -1.95741860e-15  5.01693814e-03 -1.66307302e-05  5.61933039e-04
 -1.35214641e-02  6.56434857e-02 -2.75692819e-04  2.39280689e-03
  1.86940788e-02]
Intercept: 0.005596846501107158
Fold 5 -- MSE: 0.0718, RMSE: 0.2679, MAE: 0.0908
```

In [7]:

```
# 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  0.015689  0.125256  0.071449
1     2  0.021874  0.147897  0.078476
2     3  0.012547  0.112015  0.069887
3     4  0.011545  0.107445  0.072613
4     5  0.071774  0.267906  0.090768
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js