In [1]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

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

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

In [4]:
```
# Define candidate values for alpha (λ) for Lasso Regression
alphas = [0.00392, 0.00393, 0.00394, 0.00395, 0.00396]
```

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

In [6]:
```
# To store average performance for each alpha candidate
results = []
# To store coefficients and intercepts for each fold
coefficient_results = []
```

In [7]:
```
# Loop over each candidate alpha value
for alpha in alphas:
    fold_metrics = []
    print(f"\n--- Alpha {alpha} ---")
    fold_counter = 1
    for train_index, test_index in kf.split(X):
        # Split data into train and test 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]

        # Create a pipeline: first scaling the features, then applying Lasso regression
        model = Pipeline([
            ('scaler', StandardScaler()),
            ('lasso', Lasso(alpha=alpha, max_iter=10000))
        ])

        # Train the Lasso regression model
        model.fit(X_train, y_train)

        # Extract coefficients and the intercept from the model
        coef = model.named_steps['lasso'].coef_
        intercept = model.named_steps['lasso'].intercept_

        print(f"\nFold {fold_counter} Coefficients:")
        print("Coefficients:", coef)
        print("Intercept:", intercept)

        # Store coefficient information for the current fold
        coefficient_results.append({
            'alpha': alpha,
            'fold': fold_counter,
            'coefficients': coef,
            'intercept': intercept
        })

        # Make predictions on the test set
        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({'MSE': mse, 'RMSE': rmse, 'MAE': mae})

        fold_counter += 1

    # Average metrics over the 5 folds for the current alpha
    avg_mse = np.mean([m['MSE'] for m in fold_metrics])
    avg_rmse = np.mean([m['RMSE'] for m in fold_metrics])
    avg_mae = np.mean([m['MAE'] for m in fold_metrics])

    results.append({'alpha': alpha, 'avg_MSE': avg_mse, 'avg_RMSE': avg_rmse, 'avg_MAE': avg_mae})
    print(f"\nAlpha {alpha} -- Avg MSE: {avg_mse:.4f}, Avg RMSE: {avg_rmse:.4f}, Avg MAE: {avg_mae:.4f}")
```

```
--- Alpha 0.00392 ---

Fold 1 Coefficients:
Coefficients: [ 0.01193566  0.          0.          0.01660468 -0.          0.00626665
 -0.         -0.00230171 -0.00424899  0.00454103 -0.00203568  0.00882164
 -0.        ]
Intercept: 0.04288559670781893

Fold 2 Coefficients:
Coefficients: [ 0.01035196  0.          0.          0.01840813 -0.001732    0.00308755
 -0.         -0.0021666  -0.00214098  0.00305171 -0.          0.01135243
  0.        ]
Intercept: 0.044819341563786

Fold 3 Coefficients:
Coefficients: [ 0.00609366  0.          0.00165883  0.01792982 -0.          0.00409314
 -0.         -0.00167882 -0.00940134  0.00493291 -0.          0.00408763
  0.        ]
Intercept: 0.0430213698630137

Fold 4 Coefficients:
Coefficients: [ 0.00527816  0.          0.0030764   0.01882981 -0.          0.00884039
 -0.         -0.         -0.01006121  0.00315207 -0.00194534  0.00685442
  0.        ]
Intercept: 0.04924054794520548

Fold 5 Coefficients:
Coefficients: [ 0.00853421  0.00389471  0.          0.02127411 -0.          0.00223461
  0.         -0.         -0.01003145  0.00507371 -0.          0.00994989
  0.        ]
Intercept: 0.044690273972602744

Alpha 0.00392 -- Avg MSE: 0.0153, Avg RMSE: 0.1233, Avg MAE: 0.0729

--- Alpha 0.00393 ---

Fold 1 Coefficients:
Coefficients: [ 0.01192048  0.          0.          0.01659853 -0.          0.00625298
 -0.         -0.00229901 -0.0042274   0.00453259 -0.00202408  0.0088098
 -0.        ]
Intercept: 0.04288559670781893

Fold 2 Coefficients:
Coefficients: [ 0.01033735  0.          0.          0.01839986 -0.00172037  0.00307595
 -0.         -0.00216344 -0.00212277  0.00303939 -0.          0.01134077
  0.        ]
Intercept: 0.044819341563786

Fold 3 Coefficients:
Coefficients: [ 0.00608388  0.          0.00164808  0.01792139 -0.          0.00408158
 -0.         -0.00167439 -0.00938129  0.00492026 -0.          0.00407371
  0.        ]
Intercept: 0.0430213698630137

Fold 4 Coefficients:
Coefficients: [ 0.0052649   0.          0.00306816  0.01882409 -0.          0.00882699
 -0.         -0.         -0.01003925  0.00314289 -0.00193334  0.00684442
  0.        ]
Intercept: 0.04924054794520548

Fold 5 Coefficients:
Coefficients: [ 0.0085226   0.00388743  0.          0.02126984 -0.          0.00222107
  0.         -0.         -0.01001002  0.00506107 -0.          0.00993914
  0.        ]
Intercept: 0.044690273972602744

Alpha 0.00393 -- Avg MSE: 0.0153, Avg RMSE: 0.1233, Avg MAE: 0.0729

--- Alpha 0.00394 ---

Fold 1 Coefficients:
Coefficients: [ 0.01190529  0.          0.          0.01659238 -0.          0.00623932
 -0.         -0.00229631 -0.00420581  0.00452415 -0.00201247  0.00879796
 -0.        ]
Intercept: 0.04288559670781893

Fold 2 Coefficients:
Coefficients: [ 0.01032273  0.          0.          0.01839159 -0.00170874  0.00306436
 -0.         -0.00216028 -0.00210456  0.00302707 -0.          0.01132912
  0.        ]
Intercept: 0.044819341563786

Fold 3 Coefficients:
```

```
Coefficients: [ 0.0060741   0.          0.00163733  0.01791295 -0.          0.00407002
 -0.         -0.00166996 -0.00936124  0.0049076  -0.          0.00405979
  0.        ]
Intercept: 0.0430213698630137

Fold 4 Coefficients:
Coefficients: [ 0.00525164  0.          0.00305992  0.01881837 -0.          0.00881359
 -0.         -0.         -0.01001729  0.00313372 -0.00192134  0.00683442
  0.        ]
Intercept: 0.04924054794520548

Fold 5 Coefficients:
Coefficients: [ 0.00851099  0.00388015  0.          0.02126556 -0.          0.00220752
  0.         -0.         -0.0099886   0.00504844 -0.          0.0099284
  0.        ]
Intercept: 0.044690273972602744

Alpha 0.00394 -- Avg MSE: 0.0153, Avg RMSE: 0.1233, Avg MAE: 0.0729

--- Alpha 0.00395 ---

Fold 1 Coefficients:
Coefficients: [ 0.0118901   0.          0.          0.01658623 -0.          0.00622566
 -0.         -0.00229362 -0.00418422  0.00451572 -0.00200086  0.00878612
 -0.        ]
Intercept: 0.04288559670781893

Fold 2 Coefficients:
Coefficients: [ 0.01030812  0.          0.          0.01838333 -0.0016971   0.00305276
 -0.         -0.00215713 -0.00208636  0.00301476 -0.          0.01131747
  0.        ]
Intercept: 0.044819341563786

Fold 3 Coefficients:
Coefficients: [ 0.00606431  0.          0.00162658  0.01790452 -0.          0.00405845
 -0.         -0.00166553 -0.00934119  0.00489494 -0.          0.00404587
  0.        ]
Intercept: 0.0430213698630137

Fold 4 Coefficients:
Coefficients: [ 0.00523838  0.          0.00305169  0.01881265 -0.          0.00880019
 -0.         -0.         -0.00999534  0.00312455 -0.00190935  0.00682443
  0.        ]
Intercept: 0.04924054794520548

Fold 5 Coefficients:
Coefficients: [ 0.00849939  0.00387286  0.          0.02126129 -0.          0.00219398
  0.         -0.         -0.00996718  0.0050358  -0.          0.00991765
  0.        ]
Intercept: 0.044690273972602744

Alpha 0.00395 -- Avg MSE: 0.0153, Avg RMSE: 0.1233, Avg MAE: 0.0729

--- Alpha 0.00396 ---

Fold 1 Coefficients:
Coefficients: [ 0.01187492  0.          0.          0.01658008 -0.          0.00621199
 -0.         -0.00229092 -0.00416262  0.00450728 -0.00198925  0.00877429
 -0.        ]
Intercept: 0.04288559670781893

Fold 2 Coefficients:
Coefficients: [ 0.0102935   0.          0.          0.01837506 -0.00168547  0.00304117
 -0.         -0.00215397 -0.00206815  0.00300244 -0.          0.01130581
  0.        ]
Intercept: 0.044819341563786

Fold 3 Coefficients:
Coefficients: [ 0.00605453  0.          0.00161583  0.01789609 -0.          0.00404689
 -0.         -0.0016611  -0.00932114  0.00488228 -0.          0.00403196
  0.        ]
Intercept: 0.0430213698630137

Fold 4 Coefficients:
Coefficients: [ 0.00522512  0.          0.00304345  0.01880693 -0.          0.00878679
 -0.         -0.         -0.00997338  0.00311537 -0.00189735  0.00681443
  0.        ]
Intercept: 0.04924054794520548

Fold 5 Coefficients:
Coefficients: [ 0.00848708  0.00386586  0.          0.02125733 -0.          0.00218018
  0.         -0.         -0.00994549  0.00502306 -0.          0.00990686
  0.        ]
```

```
Intercept: 0.044690273972602744

Alpha 0.00396 -- Avg MSE: 0.0153, Avg RMSE: 0.1233, Avg MAE: 0.0729
```

In [8]:
```python
# Summarize the cross-validation performance for each candidate alpha in a DataFrame
results_df = pd.DataFrame(results)
print("\nOverall Cross-Validation Results:")
print(results_df)
```

```
Overall Cross-Validation Results:
     alpha   avg_MSE  avg_RMSE   avg_MAE
0  0.00392  0.015325  0.123307  0.072916
1  0.00393  0.015325  0.123307  0.072918
2  0.00394  0.015325  0.123308  0.072921
3  0.00395  0.015325  0.123308  0.072924
4  0.00396  0.015325  0.123308  0.072926
```

In [9]:
```python
# Identify the best alpha based on average MSE
best_alpha = results_df.loc[results_df['avg_MSE'].idxmin()]['alpha']
print("\nBest alpha based on average MSE:", best_alpha)
```

```
Best alpha based on average MSE: 0.00395
```