Project Report

ADMM and Lasso Regression

Introduction

Data science and machine learning rely greatly on efficient and interpretable data analysis methods, particularly in a high-dimensional scenario. Sometimes, traditional regression fails if overfitting or a computationally costly model is suspected. Thanks to Lasso regression, researchers and practitioners could reduce dimensions and retain predictive accuracy in such applications. The core of Lasso regression is in minimizing a loss function with some added norm penalty, which forces some coefficients towards zero.

The practice of Lasso regression is studied here by making use of the Alternating Direction Method of Multipliers (ADMM) which is particularly apt for high dimensional problems since it can easily break down complex problems into several sub-problems efficiently. ADMM leads to efficient convergence across all the iterations by breaking the optimization process in the Lasso over primal and dual variables, even when optimized separately on large datasets.

The paper gives an overview of the theoretical background of Lasso regression, explains the optimization procedure using ADMM, and introduces the application of this in real-world problems such as feature selection in medical data, finance, and image processing. It defines the combination of the technique developed through the Lasso technique with the ADMM optimization procedure for better insights into its performance over various datasets and a presentation of its strengths and weaknesses compared to other optimization techniques. This work thus demonstrates the potential of Lasso regression as a sparse modeling tool and nicely illustrates how well ADMM can support algorithmic computational needs in data-intensive, complex applications.

Methodology

Application 1: Cement Strength Prediction Using ADMM Distributed Lasso

This project predicts concrete compressive strength based on mixture components and curing time. We use the ADMM (Alternating Direction Method of Multipliers) Distributed Lasso model, an optimization approach ideal for distributed computing, allowing efficient processing of large datasets.

Dataset Description

The dataset includes:

- Concrete mix components (cement, slag, fly ash, water, superplasticizer, coarse and fine aggregates),
- Curing age (days),
- Compressive strength (MPa) as the target variable.

Preprocessing steps:

1. **Data Cleaning:** Duplicates removed; features and target variable separated.
2. **Analysis:** Correlation matrix and feature distributions analyzed for relationships and outliers.

Prediction Model

The goal of Lasso regression is to predict the Concrete Compressive Strength (target variable) based on the other components (predictor variables). The Lasso objective function is as follows:

$$\min_\beta \frac{1}{2N} \sum_{i=1}^{N} (y_i - X_i\beta)^2 + \lambda\|\beta\|_1$$

Where:

- y_i: Concrete compressive strength for each observation i.
- X_i: Vector of predictor variables (cement, slag, fly ash, etc.) for each observation i.
- β: Vector of coefficients for each predictor variable.
- λ: Regularization parameter, which controls the penalty for non-zero coefficients in β.

- Distributing the Lasso Problem & Consensus Constraint:

  Distribution Across Agents

  - Divide data across M agents, where each agent m has a subset (X_m, y_m)
  - Each agent computes a local β_m based on its data, allowing parallel computation.

  Enforcing Consensus

  - To create a global solution, we need $\beta_1 = \beta_2 = \cdots = \beta_M = \beta$
  - Introduce a consensus variable z to enforce for all m, ensuring coherence across agents.

ADMM STEPS-:

1. Step 1: Update β (minimizing the least-squares term)

   In this step, we minimize the objective with respect to β, while keeping z fixed.

$$\beta^{k+1} = \left(\frac{1}{N}X^T X\right)^{-1} \left(X^T y + \rho(z^k - u^k)\right)$$

2. Step 2: Update z (shrinkage step)

   In this step, we update the z-variable, which corresponds to the L1 regularization. This is where the **soft-thresholding operator** comes into play, which performs the L1 shrinkage:

$$z^{k+1} = \text{soft}(\beta^{k+1} + u^k, \lambda/\rho)$$

3. Step 3: Update the dual variable u

   Finally, the dual variable u is updated using the following rule:

$$u^{k+1} = u^k + \beta^{k+1} - z^{k+1}$$

   This step helps to enforce the constraint $z = \beta$ by adjusting the dual variable.

Application: Feature extraction and reconstruction of Signal

The aim here is to extract features and reconstruct a synthetic signal by applying Lasso regression using the Alternating Direction Method of Multipliers (ADMM) algorithm. The goal is to showcase the sparsity-enforcing capabilities of Lasso and analyze the effectiveness of feature extraction and signal reconstruction.

Mathematical modeling:

1. Lasso Regression Objective

The objective of Lasso regression is to solve the following optimization problem:

$$\min_{\beta} \left(\frac{1}{2} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1 \right)$$

Where:

- X is the matrix of input features (m samples and n features).
- y is the vector of target values.
- β is the vector of regression coefficients to be learned.
- λ is the regularization parameter.

2. ADMM Formulation

To solve this optimization problem using the Alternating Direction Method of Multipliers (ADMM), we introduce an auxiliary variable z and reformulate the problem as:

$$\min_{\beta, z} \left(\frac{1}{2} \|X\beta - y\|_2^2 + \lambda \|z\|_1 \right)$$

Subject to the constraint: β=z

Update

$$\beta^{k+1} = \left(X^T X + \rho I\right)^{-1} \left(X^T y + \rho(z^k - u^k)\right)$$

$$z^{k+1} = \text{soft-thresholding}(\beta^{k+1} + u^k, \frac{\lambda}{\rho})$$

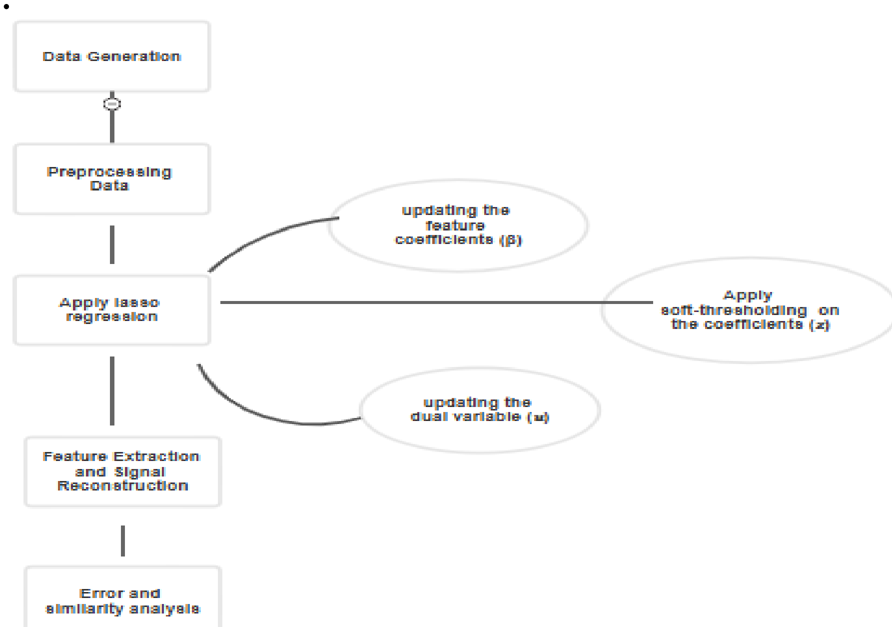
$$u^{k+1} = u^k + \beta^{k+1} - z^{k+1}$$

3. Convergence Criteria and reconstruct

$$\|\beta^{k+1} - \beta^k\|_2 < \text{tol}$$

$$\hat{y} = X\beta$$

Architecture flow:



Application: Comparison between Linear and Lasso regressions effectiveness using real life dataset

We compared lasso and linear regression by testing it on 2 datasets Predict salary using lasso and linear regression and compare results

- ***Dataset Description***

  1: Salary Dataset, It typically contains 30 rows and two columns:

  - YearsExperience: The number of years of work experience an individual has.
  - Salary: The corresponding salary for the individual

  2: Diabetes dataset,it typically has 442 rows and 11 columns

  - age: Age in years
  - sex: Gender of the patient
  - bmi: Body mass index
  - bp: Average blood pressure
  - s1: Total serum cholesterol (tc)
  - s2: Low-density lipoproteins (ldl)
  - s3: High-density lipoproteins (hdl)
  - s4: Total cholesterol / HDL (tch)
  - s5: Possibly log of serum triglycerides level (ltg)
  - s6: Blood sugar level (glu)
  - Target

- ***Data Preparation and Visualization:***
  - Exploratory Data Analysis and Visualization
  - Feature Scaling and Polynomial Transformation

- ***Model Definitions and Training***
  - Linear Regression: It solves for weights W and intercept *b* by minimizing the mean squared error (MSE) between predicted and actual values

    $$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(Y_i - (W \cdot X_i + b))^2$$

  - Lasso Regression: It extends Linear Regression by adding an L1 regularization term. This penalty encourages sparsity in the coefficients, effectively setting some of them to zero, which can improve the model's generalization by reducing complexity.

    $$\text{Lasso Objective} = \text{MSE} + \alpha\sum_{j=1}^{n}|W_j|$$

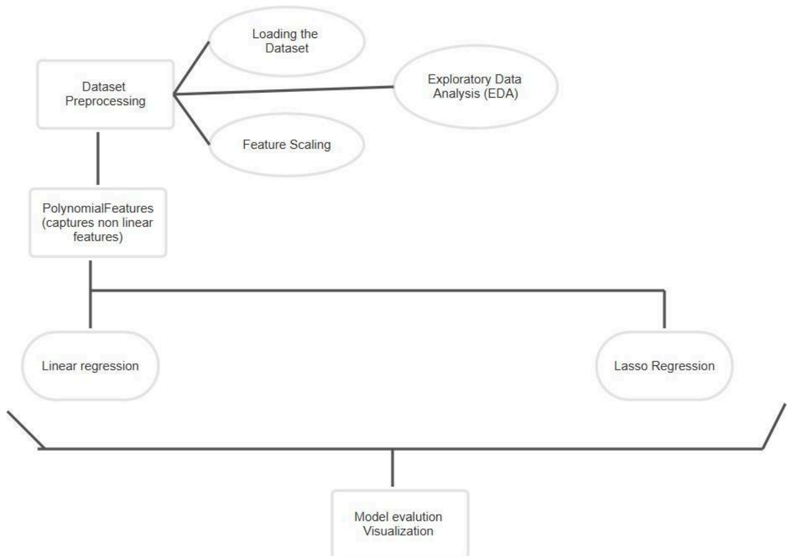
    Using ADMM:

    $$W = (X^T X + \rho I)^{-1}(X^T(Y - b) + \rho(Z - U))$$

    $$Z_j = \text{sign}(W_j + U_j) \cdot \max(|W_j + U_j| - \frac{\alpha}{\rho}, 0)$$

    $$U = U + (W - Z)$$

- System flow:



Application: Signal Processing and Reconstruction Using ADMM and Group Lasso

This application focuses on using ADMM to implement a Lasso regression function to perform Group Lasso on a given audio signal. On implementing ADMM on the given audio signal we divide it into smaller sub-divisions for the ease of applying Lasso to individual blocks and reconstruct each block and then combine them to obtain a single audio signal as the one used as input.

Dataset Description:

The dataset used here is a basic wave audio file.

Preprocessing steps:

1. Convert the given audio signal into mono audio if it is stereo audio by default
2. Determine the number of audio samples that will be processed together as a single unit, or "block" or how many samples will be present in a taken block.Calculate how many blocks will be required to cover the entire audio signal.

Mathematical Model:

Group Lasso Problem Formulation:

This reconstruction can be posed a minimizing problem for:

Here;

$$\min_{x} \frac{1}{2}\|y - \Theta x\|_2^2 + \lambda \sum_{g \in G} \|x_g\|_2$$

$$\min_{x} \frac{1}{2}\|y - \Theta z\|_2^2 + \lambda \sum_{g \in G} \|z_g\|_2$$

- y: sampled audio data (observed, compressively sampled signal).
- Θ: sensing matrix (used to map signal to measurement space).
- x: unknown signal (reconstructed signal to be recovered).
- λ: regularization parameter controlling the group sparsity.
- G: groups of coefficients for group-wise regularization.

This reformulation allows alternating updates of x and z, solving for each variable iteratively:

- Update x: Solves a quadratic minimization problem.
- Update z: Applies soft thresholding to promote sparsity in group coefficients.
- Dual variable update: Ensures constraint z=x is satisfied.

Here, the application is solved by taking the subsets of the audio signal by the help of ADMM and minimizing their sum of squared residuals (errors) using Group Lasso and finally reconstructing the input audio signal with reduced noise by inducing sparsity among the groups.

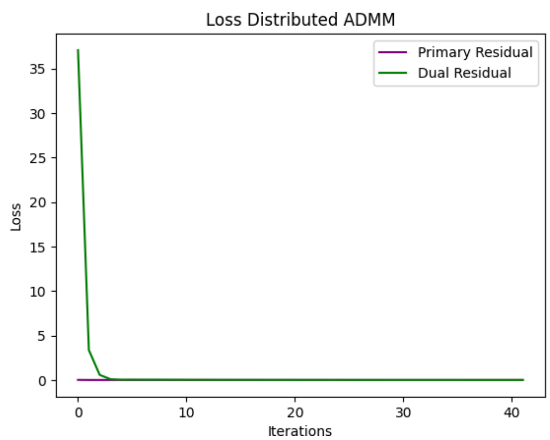
Key Mathematical Insights:

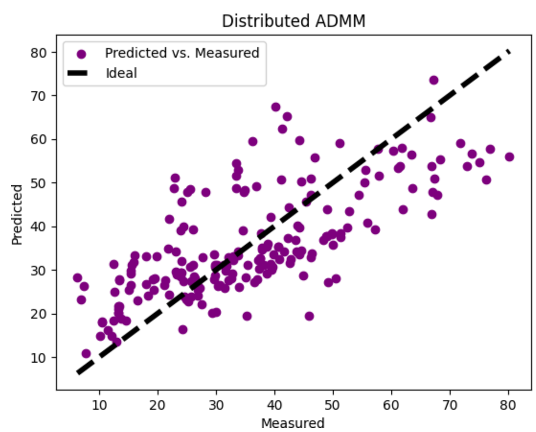
- Optimization Approach: The problem is split into manageable sub-problems using ADMM, which alternates between solving the least squares (audio fitting) and enforcing group sparsity through soft thresholding.
- Group Lasso: Encourages sparsity in groups of coefficients, leading to an effective solution for compressive sensing

Results

1. Application: Cement Strength Prediction Using ADMM Distributed Lasso

The distributed ADMM model, trained on 800 rows across 5 agents, achieved an R^2 of 0.51 with an MSE of 134.55 and MAE of 9.37. It converged in 42 iterations within 0.0188 seconds, showing efficient parallel processing and moderate predictive accuracy.





2. Application: Feature extraction and reconstruction of Signal

The output includes the learned coefficients, the reconstructed signal, and performance metrics. The reconstruction error is calculated using MSE, and the similarity between the original and reconstructed signals is measured. These results show the model's effectiveness in signal reconstruction and feature selection.

3. Application: Comparison between Linear and Lasso regressions effectiveness using real life dataset

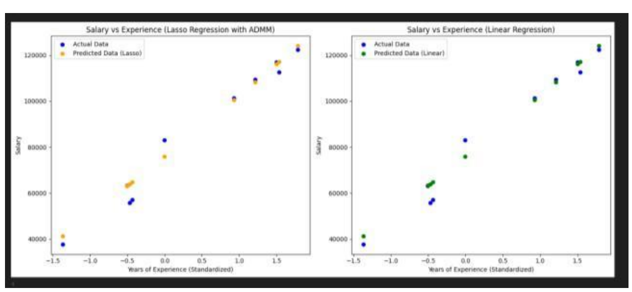
Case 1:

The EDA dataset gave us a detailed insight into a direct and proportional relationship existing between salary and experience.

The metrices compared yielded the following result pointing out that linear regression is preferred in this dataset over lasso regression

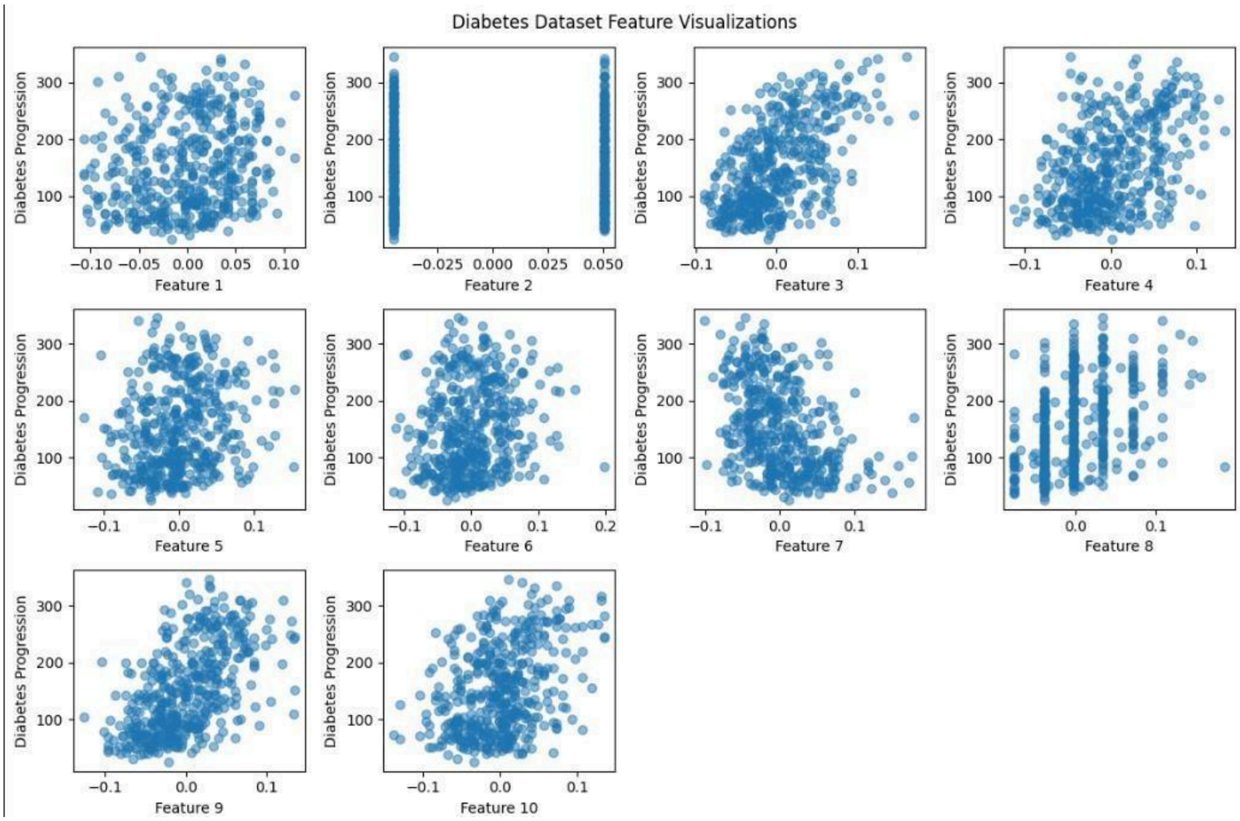
```
--- Lasso Regression (ADMM) ---
Mean Absolute Error:  3564.49
Root Mean Squared Error:  4641.9
R-squared:  0.97

--- Linear Regression ---
Mean Absolute Error:  3562.94
Root Mean Squared Error:  4643.2
R-squared:  0.97
```

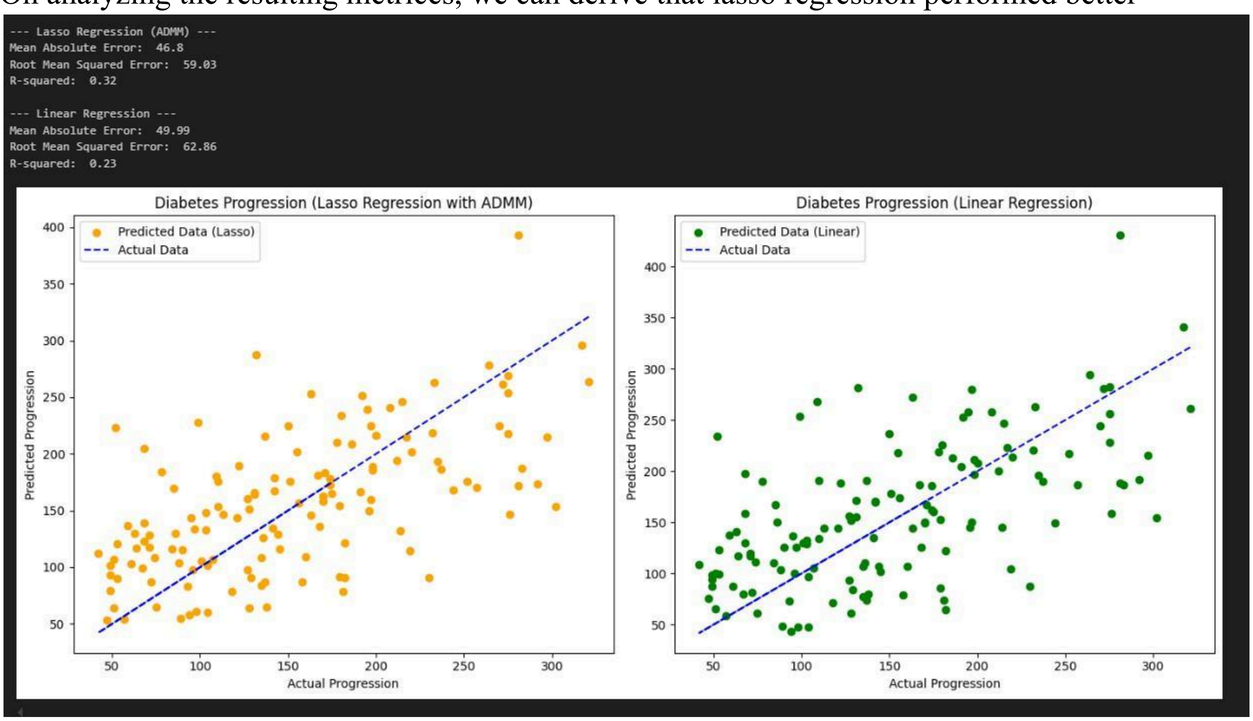


Case 2:

The EDA helps us identify that multiple factors influence results and there might be a possibility of correlation.



On analyzing the resulting metrices, we can derive that lasso regression performed better

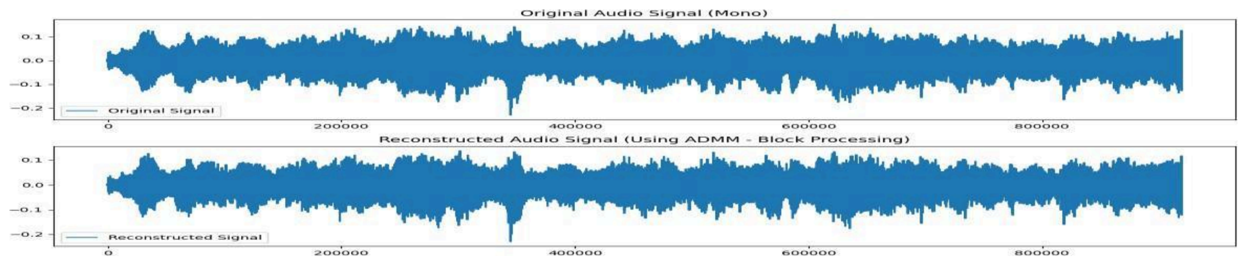


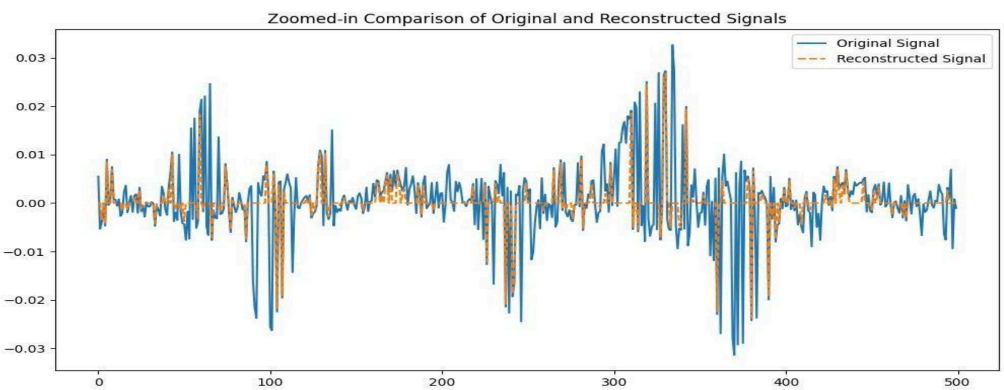
On exploring the features selected we can see clearly many features are dropped (note number of features are 66 because of Polynomial feature which sum of features, square of feature and combinations)

```
--- Important Features in Lasso Regression ---
Feature indices selected by Lasso regression: [0, 2, 3, 4, 5, 7, 9, 10, 11, 12, 14, 16, 17, 18, 19, 20, 22, 23, 26, 28, 30, 31, 32, 35, 36, 37, 38, 40, 41, 42, 43, 44, 48, 49, 52, 54, 55, 56, 58, 59, 61, 63, 65]
```

Application: Signal Processing and Reconstruction Using ADMM and Group Lasso

It can be seen that the reconstructed signal obtained is almost indistinguishable from the original signal that was started with





We have calculated the metrics such as SNR and RMSE which help quantify the amount of noise introduced during the reconstruction and the overall accuracy of the reconstructed signal.

```
SNR: 1.55 dB
RMSE: 0.0182
```

4)Application: House Sale Prediction

The aim of the project is to predict House prices using Lasso Regression using ADMM. ADMM (Alternating Direction Method of Multipliers) is a more efficient optimization method for LASSO, providing faster convergence by breaking down the problem into smaller subproblems, making it particularly effective for large-scale and distributed datasets.

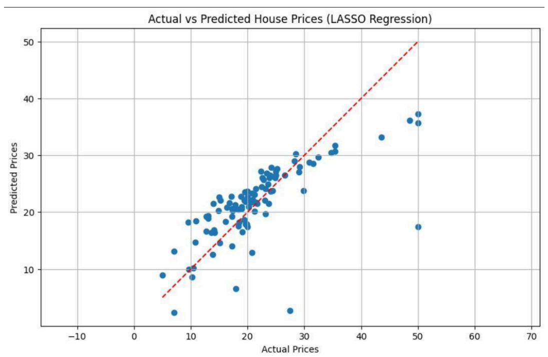
We are using a data set that contains the factors affecting the house price predictions

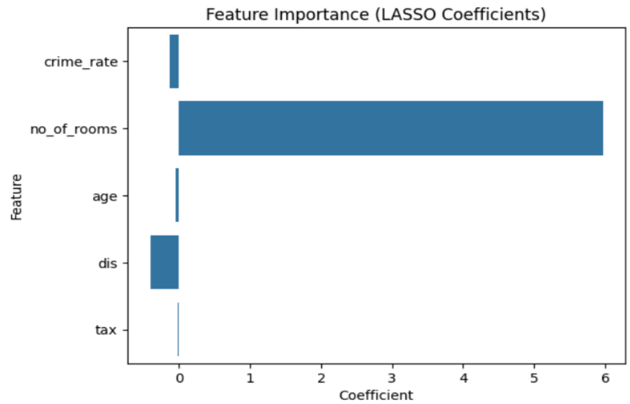
Dataset Description

- **crime_rate**: Crime rate in the town (higher values usually lower house prices).
- **zn**: Percentage of residential land with large lots (higher values often increase prices).
- **indus**: Percentage of land used for business (more industry can lower prices).
- **chas**: Whether the house is near the Charles River (1 = yes, 0 = no; being near the river increases prices).
- **nox**: Air pollution level (higher pollution usually lowers prices).
- **no_of_rooms**: Average number of rooms per house (more rooms generally increase prices).
- **age**: Percentage of old homes (older homes may reduce prices unless historically valued).
- **dis**: Distance to job centers (closer distances increase prices due to convenience).
- **rad**: Access to highways (better access may increase prices, but too close can reduce them).
- **tax**: Property tax rate (higher taxes can lower prices unless paired with better services).
- **ptratio**: Student-to-teacher ratio in schools (lower ratios suggest better schools, increasing prices).
- **b**: A measure related to the proportion of Black residents (historical biases may have influenced prices).
- **lstat**: Percentage of low-income residents (higher values often lower prices).
- **medv**: Median house price in $1000s (the value we want to predict)

We implement ADMM lasso regression function and check the functionality and efficiency by training the function using a sample dataset .After the implementation and testing of the Lasso function, we use it in a housing dataset which is used to predict house prices by looking at various factors.Lasso function trains on the dataset and reduces the non-contributing factors or constraints to zero.

We produce a plot of the predicted price and the actual price values and another plot with graph representing the significance of each coefficients according to the lasso algorithm





Future Scope

This application of group lasso with ADMM to audio data brings promising future developments across a number of fields in audio processing and machine learning. It could be tailored through the fine-tuning of parameters to an application in audio denoising and enhancement, filtering background noise while leaving intact the sounds necessary, and it might also be used to facilitate sparse coding that will be very efficient for low bandwidth streaming and storage. The model could extract features from an audio signal for use in machine learning tasks such as speech recognition or emotion detection. Group lasso can be extended to music transcription by identifying and separating instrumental sounds so that it has the potential for further applications on remixing and source separation. In healthcare and psychology, it can extract voice features associated with mood or health, which can support monitoring for mental health or speech disorders. Optimizing this model to run in real-time could open opportunities for live audio applications such as noise reduction when streaming or for a live concert scenario. Additionally, it can be combined with other data types like video to make it useful for multi-modal applications, which improves perception in autonomous systems. This project has a very good chance of becoming an all-around versatile tool applicable to consumer technology, health, and interactive media.