**Optimization I**

**Project 3: Non-Linear Programming**

***Submitted to:***

**Dr. Daniel Mitchell, Professor**

**Optimization I, R M 294**

**05240**

****

***Prepared by:***

***Krittika Deshwal (kd29275)***

***Mayank Gupta (mg66426)***

***Rahull Borana (rb47374)***

***Sanyam Jain (sj33448)***

Master of Science Business Analytics Program

**The University of Texas at Austin**

Austin, Texas

Fall 2023

**Table of Contents**

[Introduction and Scope of the Problem 3](#_Toc151147287)

[Method 1: Direct Variable Selection Method – MIQP 4](#_Toc151147288)

[Overview: 4](#_Toc151147289)

[Methodology: 5](#_Toc151147290)

[Results: 7](#_Toc151147291)

[Method 2: Indirect Feature Selection via LASSO Regression 9](#_Toc151147292)

[Overview: 9](#_Toc151147293)

[Methodology: 9](#_Toc151147294)

[Results: 11](#_Toc151147295)

[Comparison (MIQP using Gurobi vs Lasso) 12](#_Toc151147296)

[1. Comparing the errors across both methods: 12](#_Toc151147297)

[2. Comparison of the Beta Values: 13](#_Toc151147298)

[3. Predicted v/s Actual Output: 14](#_Toc151147299)

[Recommendations 15](#_Toc151147300)

[Conclusion 16](#_Toc151147301)

# Introduction and Scope of the Problem

This project compares variable selection in regression using MIQP (Mixed Integer Quadratic Programming) and LASSO methods. MIQP enforces variable inclusion through optimization, while LASSO minimizes the sum of squared residuals with an added penalty which may shrink coefficients, possibly even to zero. The objective is to assess if LASSO's shrinkage technique outperforms MIQP's direct selection method. Cross-validation helps optimize the respective hyperparameters (k for MIQP and 𝜆 for LASSO).

MIQP problems are typically very demanding computationally. Here we attempt to solve a MIQP problem using an advanced optimization software Gurobi in Python.

To implement Lasso regression, we use our standard scikit-learn which features the algorithm for Lasso in Python.

Comparing the two approaches:

#### MIQP (Mixed Integer Quadratic Programming):

* *Parameter Selection***:** Direct variable selection via optimization and binary variables.
* *Model Complexity Control***:** Determines variable inclusion, impacting model complexity.
* *Interpretability***:** May result in more complex models with a larger number of variables.

#### LASSO (Least Absolute Shrinkage and Selection Operator):

* *Parameter Selection***:** Penalizes coefficients, forcing some to zero for feature selection.
* *Model Complexity Control***:** Encourages fewer predictors with non-zero coefficients for simplicity.
* *Interpretability***:** Favors sparsity, leading to more interpretable models with fewer variables.

# Method 1: Direct Variable Selection Method – MIQP

## Overview:

The objective function, decision variables and constraints to formulate the MIQP model are as follows:

*Objective Function:*

MIQP employs a nonlinear objective function to optimize feature selection by minimizing ordinary least squares i.e., sum of squared difference between predicted and actual value for each data point.

A black text with a plus and a plus

Description automatically generated with medium confidence

Here β₀ is the intercept and β₁ - βₘ are the coefficient of the independent variables x1-xm in the dataset.

*Decision Variables:*

There are 'm' independent variables (X) and one dependent variable (Y) in the data. We consider (2m+1) decision variables in total-

1. m+1 beta values (including intercept and coefficients) - β₀, β₁, ..., βₘ

2. m binary variables - z₁, z₂, ..., zₘ

Binary variables help in selecting ‘m’ independent variables for the equation as they decide the beta coefficients to be zero or non-zero using the big-M method.

*Constraints:*

There are 2m constraints related to β and z values:

1. 
2. A black and white math symbols

   Description automatically generated with medium confidence

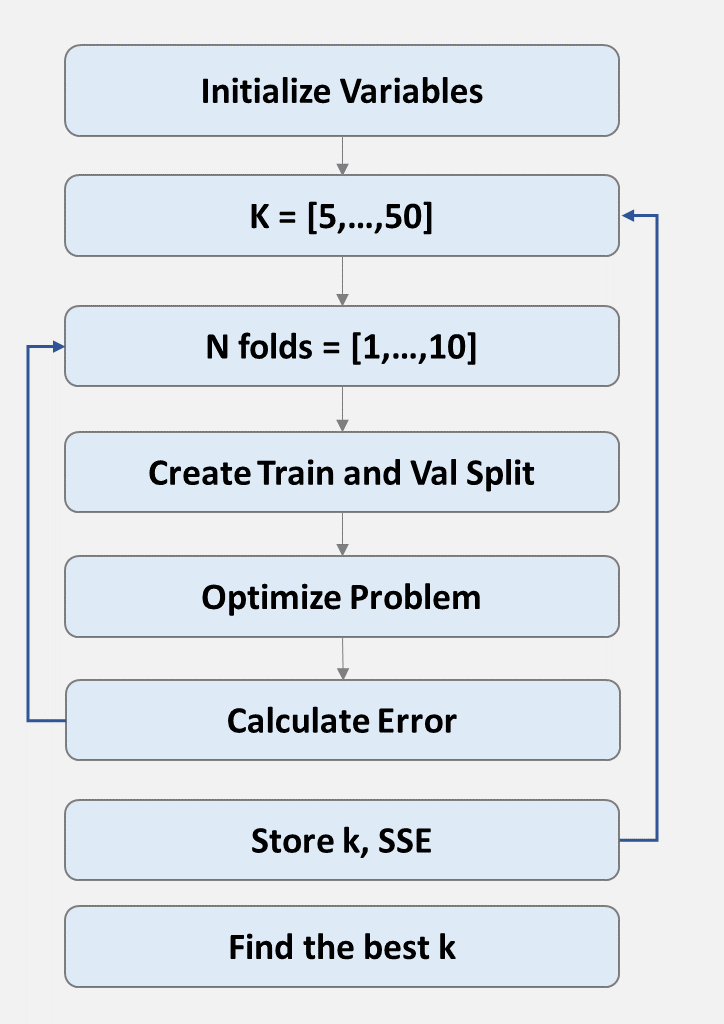
A hyperparameter 'k' is used to determine the maximum number of features to select.

First ‘m’ constraints are the big-M constraints which force beta variables to be zero or non-zero. An important point to note is that we don’t impose the constraint on the intercept variable (β₀) to become zero. The second set of constraints are for the binary variables ‘z’ where we want only ‘k’ z variables to be non-zero.

## Methodology:

To determine the best value of k, we conducted 10-fold cross-validation on the training dataset. The optimal k was chosen based on the lowest cross-validation error, which is the mean error across all 10 folds for each k value.

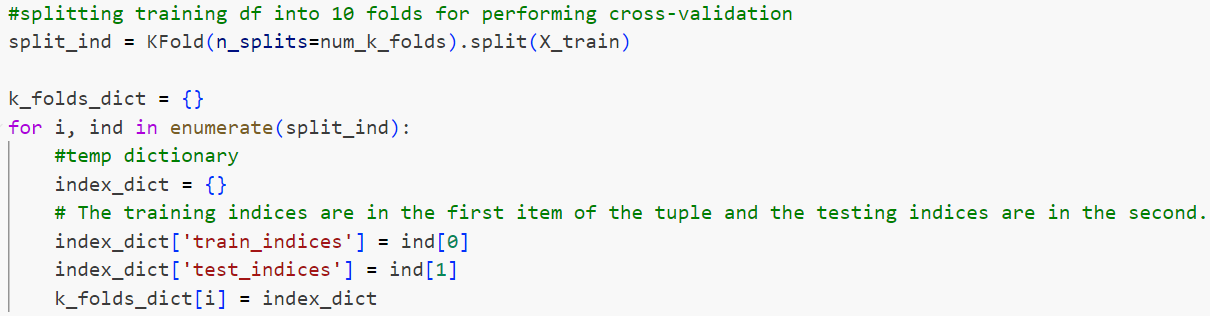
The process flow is illustrated in the flowchart below.



We applied the MIQP method on k values of 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50.



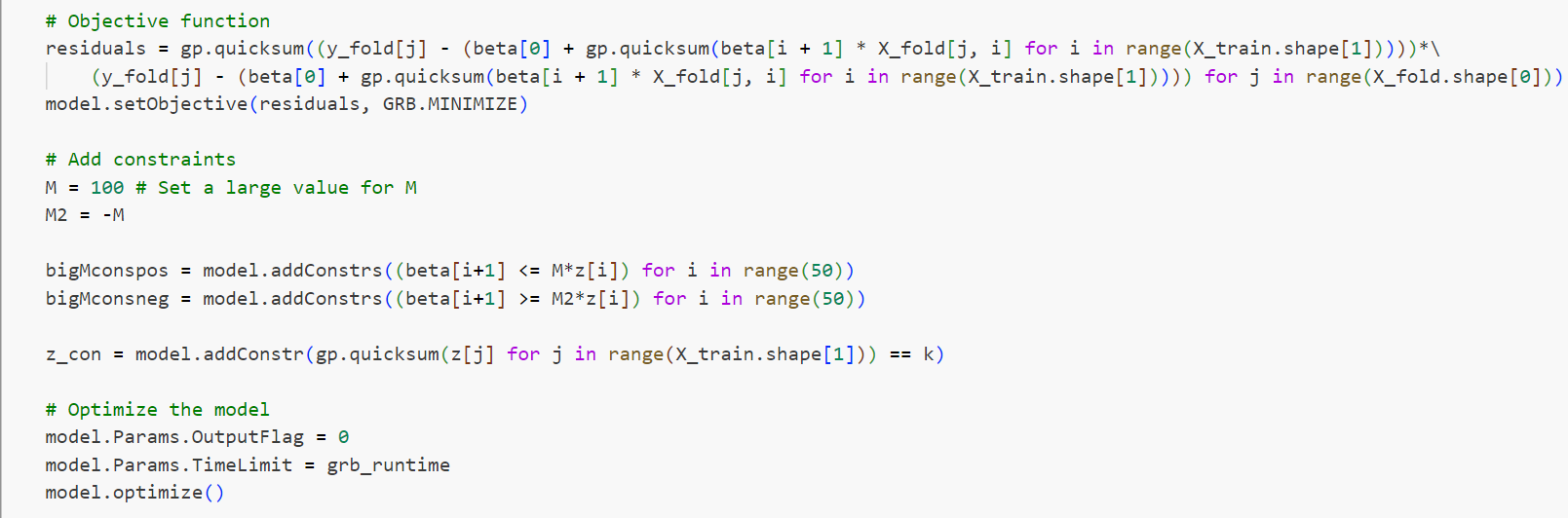
The data was divided into 10 distinct subsets of training and testing data for each fold.



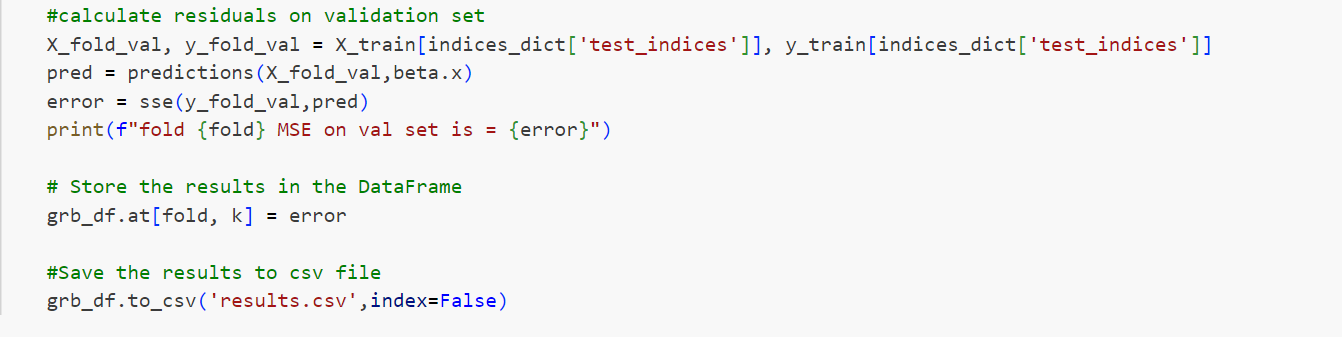
For each value of k, the data, Gurobi model and decision variables are initialized.



Objective Function and constraints were added and subsequently the MIQP model was optimized.

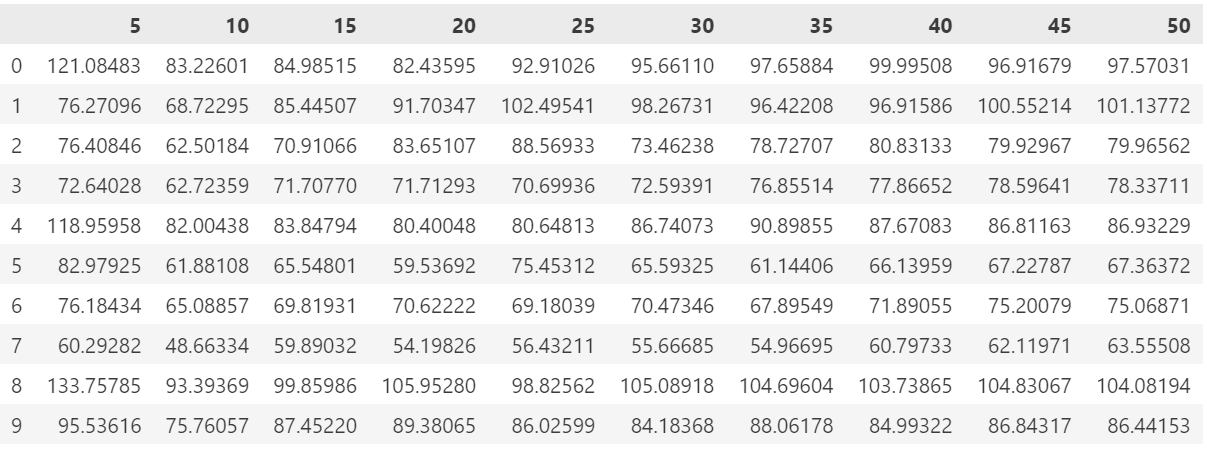


The resulting sum of squared error (SSE) was recorded in a dataframe for all values of k for each fold. The dataframe snippet is attached in the results section.

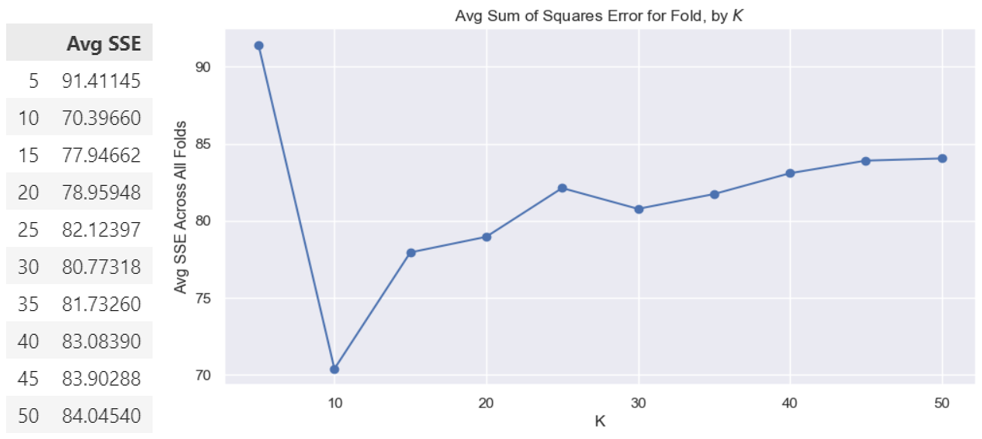


## Results:

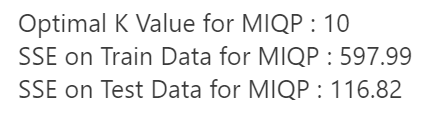
The values of Sum of Squared errors for every value of k for each fold is given below.



By averaging the SSE for each k across all 10 folds, we identified the k value that yielded the minimum cross-validation error. In this case, **k=10** was selected as the optimal value for the MIQP model as illustrated below.



The model was applied to predict the target variable in the test dataset, and its performance was assessed using the sum of squared error (SSE). The results are as follows:



# Method 2: Indirect Feature Selection via LASSO Regression

## Overview:

LASSO regression represents an indirect approach to feature selection. This method incorporates a "Shrinkage" or "Penalty" component, which effectively reduces the number of features by shrinking some coefficients to zero.

This approach adds a penalty term to the ordinary least squares method, aiding in the identification of the most effective feature combination. This technique is also referred to as L1 regularization. As the regularization parameter (λ) increases, more features are compressed towards zero, making the correct choice of λ (the penalty term) crucial. Notably, LASSO does not apply this penalty to the intercept.

*Objective Function:*

A black and white text

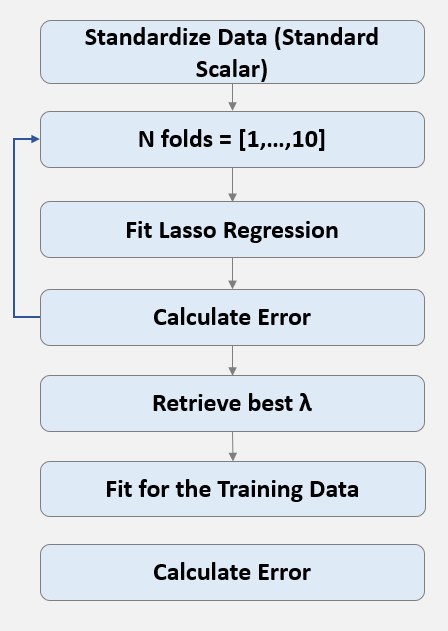
Description automatically generated

Here we have m independent variables ‘X’, a dependent variable ‘y’ and ‘n’ number of observations.

## Methodology:

The data science pipeline using Lasso regression with cross-validation involves standardizing the data, performing cross-validation with multiple folds, fitting Lasso regression models, calculating error metrics, retrieving the best regularization parameter, and evaluating the performance of the tuned model. This pipeline provides a robust approach to building and evaluating Lasso regression models for predictive tasks.

The pipeline is presented in the following flow diagram:



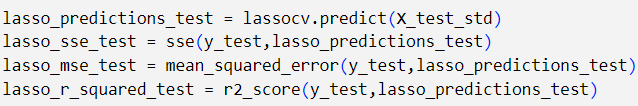
*Python Implementation:*

The LassoCV method efficiently handles the cross-validation process, providing a straightforward way to find the optimal regularization parameter for LASSO regression.

*Using LassoCv to get the best lambda and training the data with this value:*

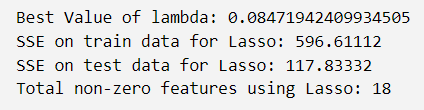


*Predict on hold out set and calculate the metrics:*



## Results:

The Lasso regression, guided by a lambda of 0.0847 from cross-validation, displayed promising test set performance with an SSE around 117.833, while the training set SSE was approximately 596.611. This suggests potential for improvement in generalization, yet the model effectively highlighted 18 essential features out of the available variables.



# Comparison (MIQP using Gurobi vs Lasso)

## Comparing the errors across both methods:

Based on the error values below, both methods seem to perform similarly. MIQP method has a slightly lower SSE compared to the LASSO method. This suggests that, on the test data, the MIQP method is performing slightly better. It is a less complex model and can generalize well with unseen data. Hence, for this dataset, MIQP performed better.

**Train results**

A close-up of numbers

Description automatically generated

**Test results**

A close-up of numbers

Description automatically generated

A graph with blue and green squares

Description automatically generated

## Comparison of the Beta Values:

A graph showing a number of values

Description automatically generated with medium confidence

LASSO identifies the optimal number of ***features as 18***, whereas ***MIQP determines it to be 10***. Additionally, it is observed that the coefficients associated with the 8 extra features identified by LASSO are notably low. MIQP excels in the selection of features that contribute significantly to explaining the variance in Y, effectively excluding features with low coefficient values.

## Predicted v/s Actual Output:

Below is the graph representing the test set predictions using both methods. We can see that predictions from both the models are comparable and track well with the target variable.

A graph showing a graph of a graph

Description automatically generated with medium confidence

# Recommendations

In the comparison presented above, LASSO opts for 18 features, resulting in a test Mean Squared Error (MSE) of 2.35. On the other hand, MIQP exhibits slightly superior performance, achieving a test MSE of 2.33 with a lower count of features. While MIQP is computationally intensive, delivering improved results, LASSO stands out for its speed and provides results that are comparable. Hence, both techniques possess their own set of advantages and disadvantages, which are elaborated as follows:

|  |  |  |
| --- | --- | --- |
|  | MIQP | Lasso |
| Advantages | a) MIQP offers a more explicit and direct method for selecting model features, providing greater control over the outcomes. | a) LASSO provides an automated system; a single line of code yields optimal values for lambda and the required number of features. |
| b) The selected features in MIQP have coefficients that are not very low, reducing variability in the model, enhancing interpretability, and improving generalization. | b) It aids in preventing overfitting by shrinking coefficients to zero. |
| c) MIQP results in a less complex model by selecting a smaller number of features. | c) Computationally efficient, LASSO is integrated into all major MLOPs platforms. |
| Disadvantages | a) It demands substantial computational power and time, particularly with large datasets or numerous features. | a) LASSO's performance diminishes when the number of features is substantially higher than the number of data points (n << p). |
| b) The output is influenced by the runtime, introducing variability in the results. | b) In the presence of correlated features, LASSO may randomly select any one of them. |
| c) It lacks automation (unlike scikit-learn), requiring manual code writing, which is time-consuming and can pose challenges for non-coders. | c) Optimal performance might result in selecting a higher number of features, making the model slightly more complex and less generalized. |
| d) The functionality is absent in MLOPs platforms. | d) Feature standardization is necessary before applying L1 penalty. |

Based on our analysis and considering the pros and cons outlined for each method, we advise against exclusively relying on one method for the company's needs. The choice of method should be contingent upon the specific use case, aligning with our business requirements and technical capabilities. It is essential to consider multiple factors when deciding on the most suitable model.

**We recommend consider using LASSO under the following conditions:**

1. Limited computational power and a quick turnaround time are crucial.
2. Frequent model runs are required, especially with regularly updated data (e.g., weekly updates), as LASSO significantly saves both time and computational resources.
3. Easy implementation is essential, as LASSO is neatly packaged and can be utilized by individuals with limited coding experience.
4. Open-source accessibility is preferred, distinguishing it from Gurobi, which requires a purchase for MIQP tasks.

**We recommend consider using LASSO under the following conditions:**

1. Abundant computational power is available, and time constraints are not a primary concern.
2. The goal is to select a predetermined number of features.
3. Explaining a substantial portion of the variance in the target variable is necessary with a minimal set of pre-specified features, particularly when acquiring additional data for extra features incurs high costs.

# Conclusion

In summary, when a model does not require frequent runs and computational power is not a limiting factor, MIQP may outperform LASSO in terms of model performance and generalization. However, it is crucial to note that MIQP comes with computational expenses and coding complexities. Given the tradeoffs involving computational efficiency, desired error margins, and coding and deployment complexity, a thorough discussion is necessary to carefully weigh these limitations and select the most suitable feature selection method aligned with the business objectives.