OPTIMIZATION
PROJECT-3

Submitted by

Chyavan Mysore Chandrashekar	CM65624
Mark Moreno	MAM24932
Parthiv Borgohain	PB25347
Soumith Reddy Palreddy	SP52466

1. Introduction

Variable selection is one of the vital steps in developing any Machine Learning algorithm. In parametric models such as regression, we have the issue of high uncertainty in the parameters obtained due to multicollinearity. In non-parametric models such as decision trees, we have the problem of loss of interpretability if there is high multicollinearity between the independent variables. Ignoring a vital variable also results in a highly biased model that can't be bypassed by acquiring more data.

Knowing the importance of this operation, we need to devise approaches that will help select the appropriate variables to build our Machine Learning models. Variable selection can be implicit or explicit. For instance, decision trees have an implicit variable selection built into the model based on how it works. It can choose not to split the tree based on a variable at any level and hence might completely ignore the independent variable. In linear regression, we can perform the explicit variable selection by employing algorithms like Forward, Backward, or Stepwise Regression. We can also use methods like LASSO regression that, in addition to being a technique used to avoid overfitting, also help in variable selection. These methods are computationally more efficient than building all possible models and then choosing the best fit model to determine which variables were most significant in the predictive task. Further, these methods are greedy and may result in a suboptimal solution, unlike the optimization approach that always leads to the optimal variables to be used in the model.

In this report, our goal is to find out which method performs better for variable selection by performing direct comparisons on the same dataset. We are going to outline the pros and cons of each method so that stakeholders can choose the model as per their needs.

2. Methodology

In this project, we plan to study two variable selection methods and discover insights to make a business recommendation of what algorithm a business can use in the future to perform variable selection based on their advantages and disadvantages.

2.1 Mixed Integer Quadratic Programming

The first approach is to use Mixed Integer Quadratic Programming (MIQP). In this method, we perform direct variable selection through optimization based on the Ordinary Least Squares (OLS) linear regression, where we use the loss function of the sum of squared errors (SSE) as our objective and try to minimize it.

2.1.1 Algorithm

We can pose the objective of the approach as the following optimization problem.

1. **Objective** - For the objective function of the model, we need to minimize the OLS SSE of the model to find the appropriate βs (coefficients). Let us assume we have m variables in the model.

$$\min_{\beta} \sum_{i=1}^{n} (\hat{y} - y)^2$$

$$\text{or}$$

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m - y)^2$$

2. **Constraints**
    a. **Variable selection constraint** - This constraint ensures only k variables are used in the regression model (the sum of all the variables present in the model should be equal to k)

$$\sum_{j=1}^{m} z_j \leq k$$

    b. **Big-M constraint** - This is a logical constraint used to force all βs to be zero unless they are selected as a variable in the model (from the previous constraint)

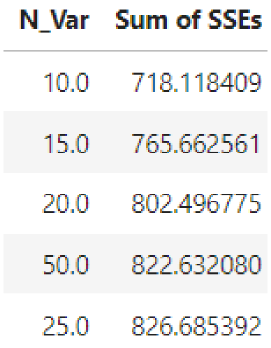
$$-Mz_j \leq \beta_j \leq Mz_j \qquad \text{for j=1,2,...m}$$

We perform 10-fold cross-validation for values of k ranging from 5 to 50 in steps of 5 to determine the optimal value of k.

After finding the optimal k, we used this k to fit the MIQP model on the training dataset. For performance evaluation, we calculated the Sum of Squared Errors (SSE) on both training and test datasets. We also plotted additional plots to gain a holistic understanding of the performance of this MIQP model.

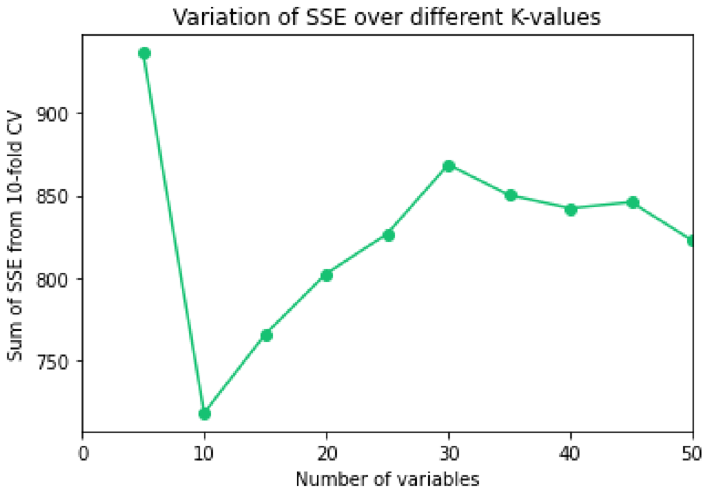
2.1.2 Results

The below table shows the 5 best results obtained in terms of SSEs after performing 10-fold cross-validation. Here N_var represents the no. of variables selected in the model.



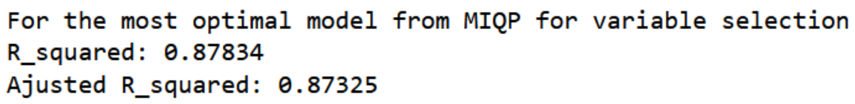
N_Var	Sum of SSEs
10.0	718.118409
15.0	765.662561
20.0	802.496775
50.0	822.632080
25.0	826.685392

The below plot shows how the SSEs vary with the number of variables chosen in the model.



Clearly, from both the table and the plot given above, we can observe that the **least SSE** is obtained for the **k** value of **10**. Thus, **k=10 gives the best fit model**.

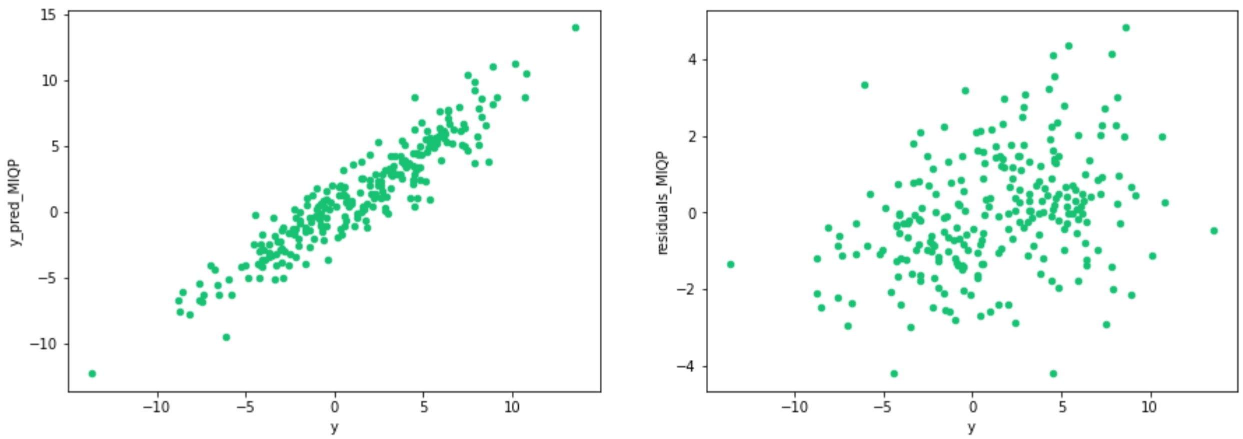
We also calculated the R-squared and Adjusted R-squared values and the results are as follows.



```
For the most optimal model from MIQP for variable selection
R_squared: 0.87834
Ajusted R_squared: 0.87325
```

It seems like a reasonable fit at an initial glance. Further, we tried to plot scatterplots to see how the model was fitting the training data.

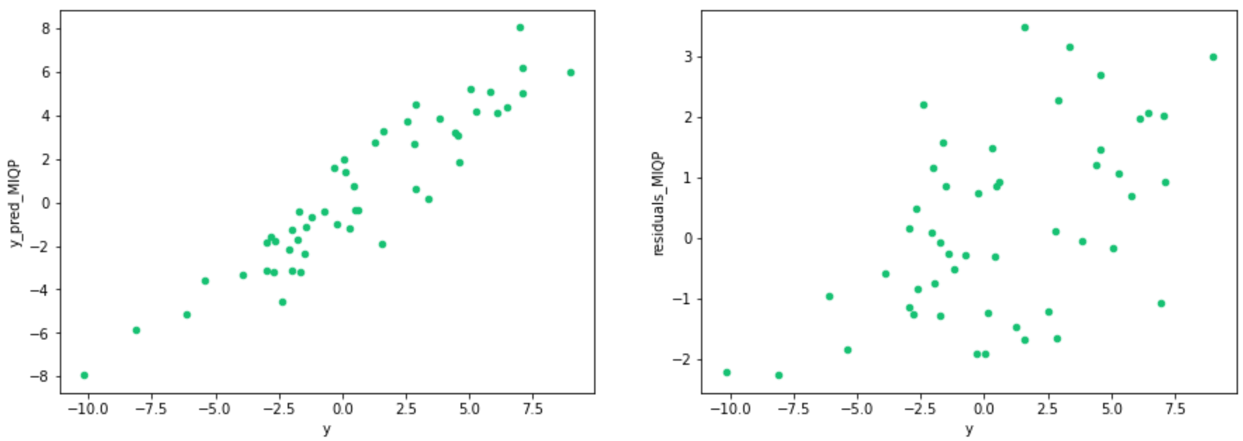
MIQP - Train: Predicted values and residuals as compared to the true 'y' values



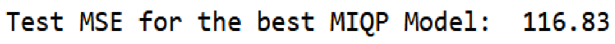
The left plot is a plot of *the predicted values of y* vs *the actual values of y*. The almost linear nature of the fit seems to suggest that this model is fitting the training data very well. The right plot is a plot of *the residuals* vs *the actual values of y*. The homoscedasticity assumption of linear regression seems to be fairly reasonable in this case looking at this plot.

Next, we tried to plot scatterplots to see how the model fit the test data.

MIQP - Test: Predicted values and residuals as compared to the true 'y' values



It seems like the model fits the test data fairly well as seen from the above plots. We also calculated the Test MSE for the best fit model. The results are shown below



```
Test MSE for the best MIQP Model:  116.83
```

2.2 Lasso Regression

The second approach - Lasso Regression - is an indirect variable selection method that regularizes the coefficients. Lasso regression is a regression method that performs regularization to enhance the prediction accuracy and interpretability and the model, by design, also performs feature selection implicitly.

2.2.1 Algorithm

We can pose the objective of the approach as the following problem.

Objective - For the objective function of the model, we need to minimize the combination of the OLS SSE term of the model in addition to the penalizing term. Our objective is thus to find the appropriate βs (coefficients) given the lambda (λ) coefficient for the regularization penalty term. The objective function can be mathematically expressed as follows

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} - y_i)^2 + \lambda \sum_{j=1}^{m} |\beta_j|$$

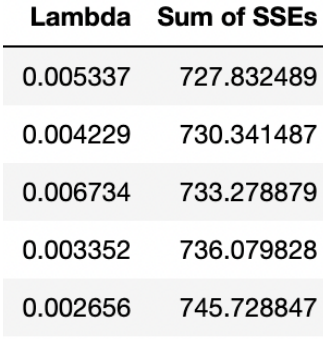
This λ hyperparameter can be determined by leveraging cross-validation. A high value of λ would drive several βs (coefficients) to zero and this also has the additional advantage of shrinking the βs which helps to address the overfitting problem. Note that if the λ value is tiny the problem would be equivalent to solving the OLS problem.

We used the sklearn package readily available in python to perform this lasso regression and determine the optimal value for the hyperparameter. We performed 10-fold cross-validation on a range of lambda values ranging from 10^{-5} to 10^{5} (in 100 steps) to determine the optimal λ hyperparameter.

After finding the optimal value of λ, we used the λ to fit the whole training data to get the best model with the optimal number of variables.

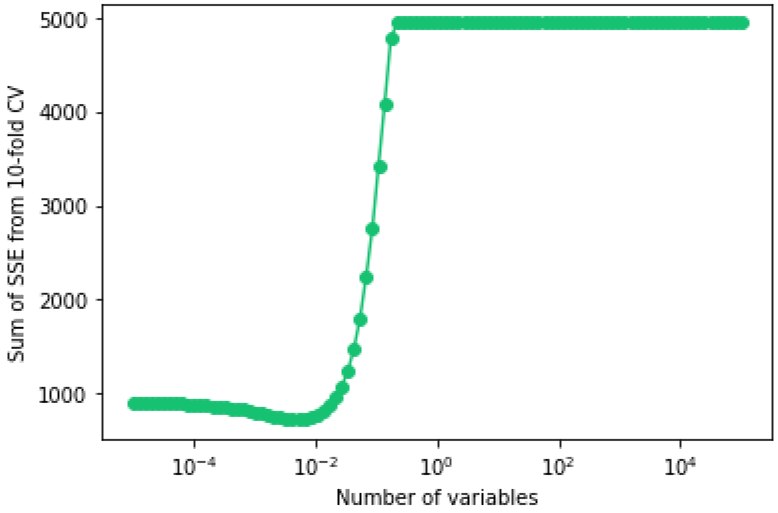
2.2.2 Results

The below table shows the 5 best results after performing 10-fold cross-validation obtained in terms of the sum of SSEs.



Lambda	Sum of SSEs
0.005337	727.832489
0.004229	730.341487
0.006734	733.278879
0.003352	736.079828
0.002656	745.728847

Here is a plot showing how the cost function varies with the number of variables selected

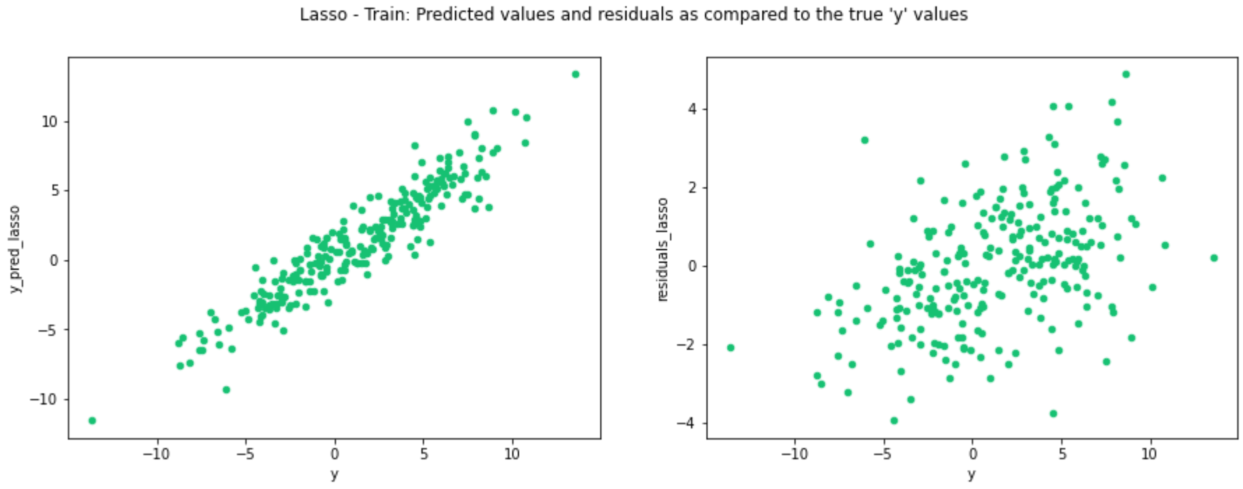


We found that the best λ value is 0.0053367 which gives the minimal error of 727.832489 with 18 features selected.

The best λ when used to fit the whole training data gives the following R-squared and Adjusted R-squared values for the model.

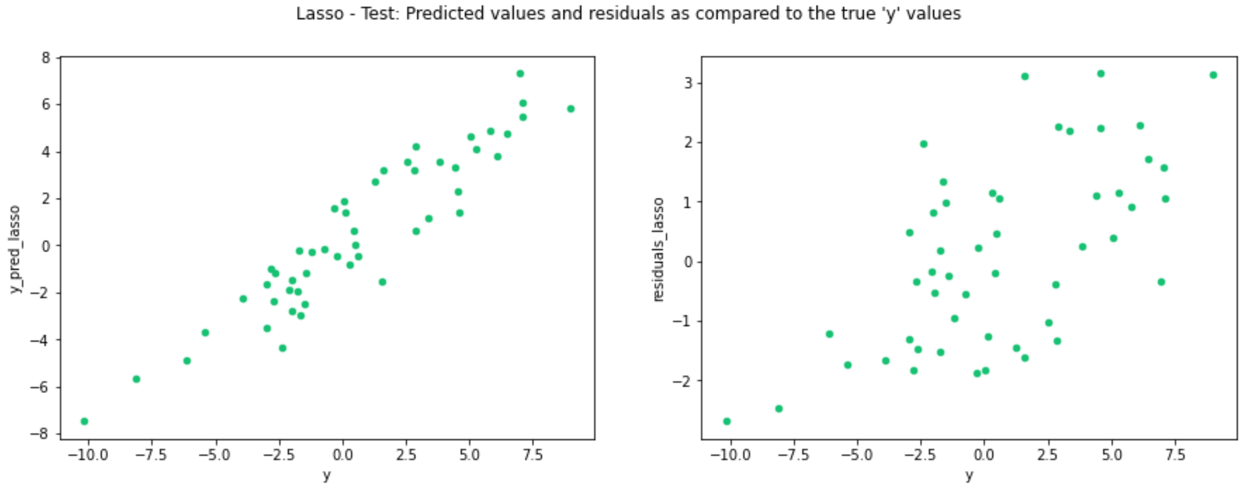
```
For the most optimal model from Lasso for variable selection
R_squared: 0.87866
Ajusted R_squared: 0.86921
```

It seems like a reasonable fit at an initial glance. Next, we tried to plot scatterplots to see how the model was fitting the training data.



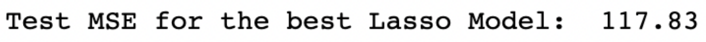
The left plot is a plot of *the predicted values of y* vs *the actual values of y*. Similar to the previous approach, the almost linear nature of the fit seems to suggest that this model is fitting the training data very well. The right plot is a plot of *the residuals* vs *the actual values of y*. The homoscedasticity assumption of linear regression seems to be fairly reasonable in this case looking at this plot.

Next, we tried to plot scatterplots to see how the model fit the test data.



We also plotted two scatter plots just like the previous ones except this time we did it for test data. In these plots too, it seems like the model is fitting the test data fairly well.

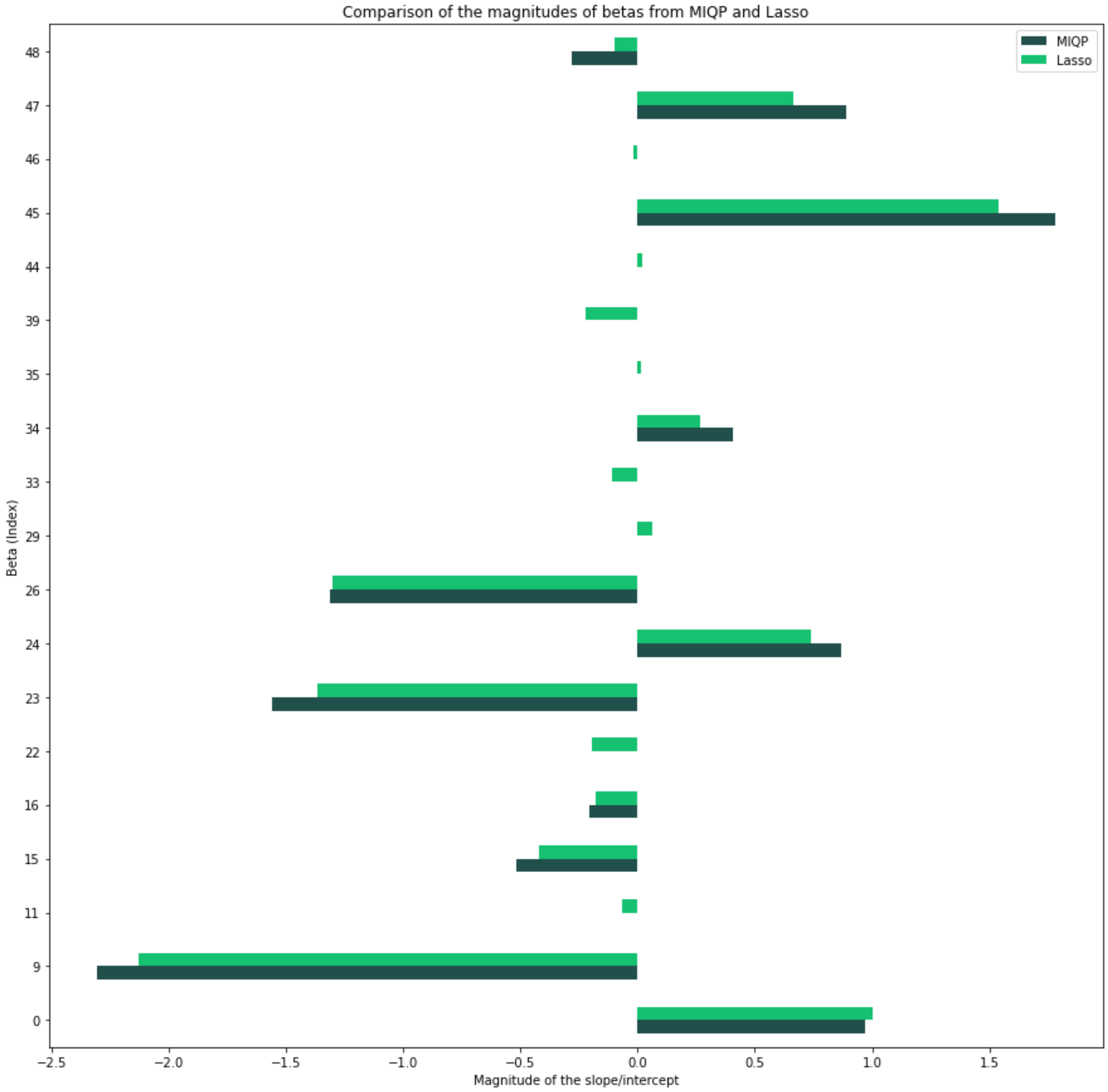
The test MSE for the best lasso model is as follows



```
Test MSE for the best Lasso Model:  117.83
```

3. Comparison

3.1 Beta Comparison



Here, we see how the magnitude of the non-zero βs from MIQP or Lasso are related from the two variable selection models. As seen in the figure above

- We can see most of the variables almost have similar βs in both models
- The non-zero βs, i.e., βs that are non-zero in Lasso but are zero in MIQP have very low β values.
- We can see that Lasso has a slightly lower magnitude for βs as compared to their counterparts in MIQP. This is expected as Lasso penalizes βs to introduce a bias.

3.2 Pros and Cons

3.2.1 Advantages of Mixed Integer Quadratic Programming

- The solution obtained from this method is guaranteed to be optimal and thus results in the best variables that can be selected through the process
- We can pre-specify the number of variables to be included in the model based on the complexity we are looking for

3.2.2 Advantages of Lasso Regression

- Faster in computation compared to MIQP. As we saw in this case
- In addition to variable selection, it also helps in introducing bias to the model to find the optimal bias-variance trade-off to get the minimal Total Error

$$Total\ Error = Bias^2 + Variance + irreducible\ error$$

3.3 Tabular Results

We have trained two types of models for variable selection to choose from **50** variables with a data of **250** observations. Let us see how the models compare.

	MIQP	**Lasso**
Number of models trained	100 *(10 Ks x 10-folds)*	1000 *(100 λs x 10-folds)*
Time taken (minutes:seconds)	31:59.83	00:06.82
Number of variables in the resulting model	10	18
R-2 Score	0.87834	0.87866
Adjusted R-2 Score	0.87325	0.86921
Test MSE	116.83	117.83

Even with higher computational power, using MIQP for variable selection seems to be a time-taking process for data with just 50 variables and 250 observations, and a ten times lesser number of models as compared to Lasso regression. We can also see that there is no significant difference when it comes to the explainability of the two models with the R-2 and Adjusted R-2 scores. Similarly, the Test MSEs also seem to have very similar values in both cases. We also saw that the βs in both models are very similar, and the additional variables included in the lasso regression do not have significant magnitudes of βs implying that changes in those variables do not significantly affect the response.

4. Business Recommendation

As shown throughout our report, variable selection plays an important role in building efficient models. When building a model we must analyze the problem at hand to determine the choice of either lasso or direct variable selection. Mixed integer quadratic programming provides more accurate results by reducing the sum of squared errors but took considerably more time than the Lasso regression which provided results that were similar but not as accurate as the mixed integer quadratic programming results.

Even though it took a significant amount of time to run, the MIQP method chose exactly the number of variables required by us. This can be helpful in scenarios where the dataset has a very high number of variables and we want to retain only a few of them such that the model is interpretable for a wider business audience. On the other hand, Lasso was much quicker in execution. Its major downside is that we cannot pre-specify the number of variables we want in the model. Lasso will do this selection on its own as a byproduct of regularization.

Our recommendation is as follows

- If the company has the resources to run mixed integer quadratic programming on a dataset, the task calls for as much accuracy as possible, and the company has enough data points for the dataset under consideration so as to not overfit the model during training, the company should choose MIQP. Also, if we know how many variables we want to include in our model beforehand, MIQP is a better approach.
- However, If we have a large dataset with many dimensions, this option may not be feasible and Lasso would be a better model to fit and use for variable selection.

All in all, both of these options are useful and both have a place in analysis. Choosing between the two depends on the individual circumstances at the company.