Final Project : Elastic Net Model

Presented to
Professor Sanjith Gopalakrishnan

By
Chaturvedi, Jaya - 261151169
Clarke, Sean - 260706014
Delisle, Audrey - 261142504
Elenany, Mohamed - 260892581

MGSC 695 Optimization for Data Science - Section 075

McGill University - Desautels Faculty of Management
Tuesday March 5th 2024

Table of Contents

Section 1 : Introduction

1.1 Brief overview of the project

In this final project for MGSC 695: Optimization for Data Science, our team has decided to look into the intricacies of machine learning through the lens of the elastic net method. Our project is designed to not only understand but also implement this method from the ground up, helping us undedrstand the underlying mechanics beyond the convenience of pre-packaged solutions. Elastic net, a regularization technique combining the strengths of Lasso (L1 regularization) and Ridge (L2 regularization) regression methods, is our main project topic. By applying this method to a real-world dataset, specifically the "Student Final Grade Prediction" dataset from the UCI Machine Learning Repository[1], we aim to predict student final grades based on a comprehensive set of features ranging from demographic to school-related attributes. This project will not only test the efficacy of our implementation but also provide a comparative analysis against black-box implementations, offering insights into the practical applications and limitations of the elastic net in the field of data science.

1.2 Purpose and goals of the project

The primary purpose of this project is to deepen our understanding of the elastic net method by taking on the challenge of implementing it from scratch. Our main goals are:

- To successfully implement the elastic net method, ensuring a comprehensive understanding of its mathematical foundation and algorithmic structure.
- To apply this implementation to the chosen dataset, thereby demonstrating its practical utility in predicting outcomes based on real-world data.
- To conduct a thorough comparison with existing black-box implementations, thereby evaluating the performance, strengths, and weaknesses of our approach.
- To write a reflection on our process and reflect on what we have learned.

1.3 Importance of the chosen machine learning method

The elastic net method represents a significant advancement in the world of machine learning, particularly in scenarios where the number of predictors surpasses the number of observations or when several predictors are highly correlated. By blending Lasso's ability to perform variable selection with Ridge's capacity to handle multicollinearity, the elastic net method offers a robust solution to regression problems. Its importance lies not only in its technical merits but also in its wide applicability across various domains, from finance to healthcare, where predictive accuracy and model interpretability are very important. Through this project, we aim to explore the elastic net's dual nature, comparing it to lasso individually, ridge individually and the black-box implementations of both these methods.

[1] Kaggle. (January 2024). Student Final Grade Prediction, https://www.kaggle.com/datasets/tejas14/student-final-grade-prediction-multi-lin-reg

Section 2 : Background Theory

2.1 Description and details of the selected machine learning method

As a team, we have decided to focus on the elastic net as our machine learning method of choice. The elastic net is a regularization and variable selection technique that combines the properties of both Lasso (Least Absolute Shrinkage and Selection Operator) and Ridge regression methods. This hybrid approach incorporates two key parameters: λ, which controls the overall strength of the regularization, and α, which balances the contribution of Lasso and Ridge penalties. Elastic net is particularly useful in situations where there are numerous correlated predictors or when the number of predictors significantly exceeds the number of observations, making it a compelling choice for our project.

2.2 Overview of how the method works

Our chosen machine learning method, Elastic Net, is a hybrid regularization technique that combines the principles of Lasso and Ridge regression. Before diving into Elastic Net, let's quickly revisit ridge and lasso. Ridge regression employs an L2 penalty, mitigating overfitting by penalizing large model parameters. On the other hand, lasso regression utilizes an L1 penalty, promoting sparsity in the model by encouraging certain parameters to be exactly zero. Elastic Net, as the name suggests, forms a nexus between ridge and lasso by seamlessly incorporating both L1 and L2 penalties into its objective function. The key innovation here lies in the flexibility it provides, offering a dual regularization approach without the need to choose between ridge or lasso. This is particularly advantageous when faced with uncertainty regarding the significance of features. The below functions were taken from an article written by Giba, L. in 2024[2]. The first function is the loss function of ridge regression, while the second one is the loss function of lasso regression. The third function is a combination of the two penalties, in order to get the loss function of elastic net.

$$\begin{aligned} RidgeMSE(y, y_{pred}) &= MSE(y, y_{pred}) + \alpha \sum_{i=1}^{m} \theta_i^2 \\ &= MSE(y, y_{pred}) + \alpha ||\boldsymbol{\theta}||_2^2 \end{aligned}$$

$$\begin{aligned} LassoMSE(y, y_{pred}) &= MSE(y, y_{pred}) + \alpha \sum_{i=1}^{m} |\theta_i| \\ &= MSE(y, y_{pred}) + \alpha ||\boldsymbol{\theta}||_1 \end{aligned}$$

$$\begin{aligned} ElasticNetMSE &= MSE(y, y_{pred}) + \alpha_1 \sum_{i=1}^{m} |\theta_i| + \alpha_2 \sum_{i=1}^{m} |\theta_i| \\ &= MSE(y, y_{pred}) + \alpha_1 ||\boldsymbol{\theta}||_1 + \alpha_2 ||\boldsymbol{\theta}||_2^2 \end{aligned}$$

[2] Giba, L. (2024). Elastic Net Regression Explained, Step by Step. Machine Learning Compass. https://machinelearningcompass.com/machine_learning_models/elastic_net_regression/

Here, α1 controls the strength of the L1 penalty, α2 controls the strength of the L2 penalty, and θi represents the model parameters. Elastic Net introduces two tuning parameters, α1 and α2, allowing practitioners to modulate the balance between L1 and L2 regularization. If α1=0, Elastic Net converges to ridge regression, while α2=0 corresponds to lasso regression. The user can also opt for a unified approach by specifying both α and an L1-ratio parameter, offering further flexibility in penalty assignment.

In our implementation, we carefully considered the impact of α on the regularization strategy. A higher α promotes sparsity, encouraging the model to rely on fewer features, similar to Lasso. Conversely, a lower α emphasizes stability, similar to Ridge. Striking the right balance between feature selection and stability is a critical aspect of tuning Elastic Net for optimal performance. Understanding the interplay between Lasso and Ridge within Elastic Net provides our group with a nuanced approach to regularization, making informed decisions about model complexity and feature importance during the iterative optimization process.

2.3 Implementation Steps

Let's look into the detailed steps of implementing Elastic Net using Gurobi. Elastic Net is a linear regression model that combines both L1 (lasso) and L2 (ridge) regularization. Our goal is to optimize the coefficients of the model by minimizing a specific objective function. Gurobi comes into play as the powerful optimization engine that navigates through the vast solution space to find the optimal coefficients. The below steps are tailored to elastic net, but they are applicable to Lasso and Ridge as well.

1. Formulating the Objective Function: We start by defining the objective function that Gurobi will minimize. In Elastic Net, this function combines the standard linear regression loss (sum of squared errors) with L1 and L2 regularization terms: .

$$\text{Minimize } \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \frac{\lambda_2}{2}\sum_{j=1}^{p} \beta_j^2$$

   Here, yi is the target variable for observation i, xij is the value of feature j for observation i, βj is the coefficient for feature j, β0 is the intercept, and λ1 and λ2 are regularization parameters controlling the strength of L1 and L2 regularization, respectively.

2. Setting Constraints: To guide Gurobi in finding meaningful coefficients, we imposed constraints on the variables. Specifically, we enforced that the absolute values of the coefficients must be greater than or equal to the actual coefficients: $|\beta j| \geq$ *Actual Coefficient j*. This introduces sparsity in the model, as some coefficients will be exactly zero, effectively performing feature selection.

3. Configuring Gurobi for Optimization:We then set up Gurobi to tackle this optimization problem. We define the decision variables, which are the coefficients βj and their absolute values. Gurobi's task is to find the optimal values for these variables.

4. Solving the Optimization Problem: Gurobi employs advanced optimization algorithms to navigate the solution space efficiently. It adjusts the coefficients and their absolute values to minimize the objective function while satisfying the imposed constraints.

5. Extracting the Optimal Coefficients: Once Gurobi completes the optimization process, we extract the optimal coefficients βj. These coefficients represent the solution to the Elastic Net problem, balancing predictive accuracy with regularization.

6. Model Evaluation: To ensure the effectiveness of our Elastic Net model, we evaluate it using test data. We make predictions based on the optimized coefficients and assess performance metrics such as the sum of squared errors and R-squared.

7. Cross-Verification with scikit-learn: To validate the correctness of Gurobi's findings, we cross-verify the coefficients obtained through Gurobi with those obtained using scikit-learn's Elastic Net implementation. This step ensures consistency and reliability in our model.

In summary, the Elastic Net implementation using Gurobi involves formulating a comprehensive objective function, setting constraints to guide the optimization, configuring Gurobi for the task, solving the optimization problem, and finally evaluating and cross-verifying the model. It's a detailed process that leverages optimization techniques to extract the most accurate coefficients for predictive modeling.

Section 3 : Data Methodology

3.1 Selected Dataset

The dataset chosen for this project is titled "Student Final Grade Prediction-Multi_lin_reg," sourced from the UCI Machine Learning Repository[3]. The dataset revolves around the academic performance of secondary education students from two Portuguese schools. It encompasses a variety of attributes, including student grades, demographic information, and social and school-related features. The dataset provides insights into the performance of students in two distinct subjects: Mathematics (mat) and Portuguese language (por). It is noteworthy that the target attribute, G3, exhibits a strong correlation with attributes G2 and G1. This correlation arises because G3 represents the final year grade issued at the 3rd period, while G1 and G2 correspond to the 1st and 2nd period grades, respectively. Predicting G3 without considering G2 and G1 is more challenging but significantly more valuable, as explained in the source description.

The selection of the "Student Final Grade Prediction-Multi_lin_reg" dataset stemmed from our criteria of finding a dataset that aligns with ease of implementation for regression models and contains predominantly numerical values. This dataset, with its focus on predicting students' final grades in Portuguese secondary schools, not only meets the technical requirements for regression analysis but also captivates our interest in contributing to the educational domain. As students ourselves, we were particularly drawn to a dataset that resonated with our educational background and could potentially have real-life implications. The idea of exploring factors influencing academic performance and, by extension, assisting the education industry in making informed decisions underscores the practical significance of this dataset. Its relevance to our own experiences as students adds an extra layer of motivation for our analytical research.

3.2 Data Preprocessing

In the process of preparing our dataset for predictive modeling, we begin by a preliminary examination of the dataset involving checking for missing values and duplicate rows, ensuring data integrity. The subsequent preprocessing steps include dropping columns deemed unnecessary for our prediction task, such as school-related details, demographic information, and reasons for choosing a school. The remaining dataset is then subjected to one-hot encoding, a technique used to convert categorical features into a numeric format. This process facilitates the inclusion of categorical data in machine learning models, enhancing the overall predictive capability. Further, we split the dataset into features ('X') and the target variable ('y'), where 'X' represents the independent variables, and 'y' corresponds to the final year grades ('G3'). To streamline our dataset for training and evaluation, we split it into training and test sets using the 'train_test_split' function from scikit-learn. The training set, constituting 80% of the data, serves to train our models, while the test set (20% of the data) allows us to assess the model's performance on unseen data. Lastly, we used StandardScaler from scikit-learn to standardize our features.

[3] Kaggle. (January 2024). Student Final Grade Prediction, https://www.kaggle.com/datasets/tejas14/student-final-grade-prediction-multi-lin-reg

Section 4 : Experimental Setup

4.1 Lasso Procedure

In the LASSO experimental setup, we used both scikit-learn and Gurobi optimization software for implementation. The scikit-learn implementation involved a grid search over a range of alpha values to identify the optimal regularization strength. The selected alpha value (0.0655), along with its corresponding performance metrics (negative mean squared error, MSE, and R-squared), provided insights into the model's effectiveness. The resulting LASSO model demonstrated predictive performance with a sparsity-inducing effect on the coefficients.

The Gurobi implementation tackled the LASSO problem using quadratic programming. Due to the nonlinearity introduced by the L1-norm term, the formulation required a clever introduction of auxiliary variables and constraints. These transformed the problem into a quadratic programming (QP) problem that Gurobi could efficiently solve. The code notation is below:

Solution Formulation - Lasso Linear Regression:

$$\min_{\beta} \sum_{i=1}^{n}(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_1 x_{ip} - y_i)^2 + \sum_{j=1}^{m} \lambda_j |\beta_j|$$

Matrix notation:

$$\min_{\beta} \beta^T(\mathbf{X}^T\mathbf{X})\beta - 2\beta^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y} + \lambda|\beta|$$

where:

λ = lambda vector
$\lambda_1 = 0$, the penalization doesn´t affect the intercept
$\lambda_{j\neq 1} = l_1$ penalization (alpha)

However, it's difficult to solve the former formulation since that QP problem doesn´t meet the same form that the General QP formula because $|\beta|$ is a not linear and we cannot a proper form for **c**. Hence, we reformulated the problem into a quadratic problem.

Reformulation for Quadratic Programming

To solve this problem using Gurobi, a quadratic programming solver, we introduce auxiliary variables and constraints to model the L1-norm term:

Variables

- Variables β_{pos} and β_{neg} for each coefficient to represent the positive and negative parts, respectively.
- z_i binary variables to model the selection of β_{pos} or β_{neg}.

Objective Function (Reformulated)

$$\min_{\beta_{\text{pos}},\beta_{\text{neg}}} \frac{1}{2n}\|y - X(\beta_{\text{pos}} - \beta_{\text{neg}})\|_2^2 + \alpha \sum(\beta_{\text{pos}} + \beta_{\text{neg}})$$

Constraints

To ensure that either β_{pos} or β_{neg} is selected but not both, introduce constraints:

$$\beta_{\text{pos},i} \leq M z_i$$

and

$$\beta_{\text{neg},i} \leq M(1 - z_i)$$

where M is a sufficiently large number, and i indexes over the coefficients.

Implementation Notes

In the Gurobi model:

- Q and c represent the quadratic and linear parts of the objective function, respectively.
- A, b, and $sense$ represent the constraints that link the original β variables with the auxiliary β_{pos}, β_{neg}, and z_i variables.
- The final solution for β is obtained by subtracting β_{neg} from β_{pos}, and the intercept is handled separately if included.

4.2 Ridge Procedure

Similar to lasso, in the Ridge experimental setup, we used both scikit-learn and Gurobi optimization software to implement the Ridge regression model. The scikit-learn implementation involved a grid search over a range of alpha values, representing the regularization strength. The optimal alpha value (1.4563) was identified through cross-validation, and its corresponding performance metrics, including mean squared error (MSE) and R-squared, were carefully evaluated on both the training and test datasets.

For the Gurobi implementation, the Ridge procedure was formulated as a quadratic programming (QP) problem. The objective was to minimize the sum of squared errors (SSE) between predicted and actual values while incorporating L2-norm regularization to prevent overfitting. This was achieved by introducing a regularization term that penalizes large coefficient values. The Gurobi model was configured with appropriate variables and constraints to efficiently solve this optimization problem. The code notation is below:

Solution Formulation - Ridge Linear Regression:

Objective: Minimize the sum of squared errors (SSE) between the predicted values and the actual target values, while also penalizing large coefficient values to prevent overfitting.

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_1 x_{ip} - y_i)^2 + \sum_{j=1}^{m} \lambda_j \beta_j^2$$

The objective function consists of two parts:

1. Sum of Squared Errors (SSE): This term measures the discrepancy between the predicted values and the actual target values.
2. Regularization Term: This term penalizes large coefficient values to prevent overfitting. It is the sum of the squared coefficients multiplied by the regularization parameter lambda.

Sum of Squared Errors (SSE): In the context of matrix notation, this term can be represented as the squared Euclidean norm of the difference between the predicted values and the actual target values. Regularization Term: The regularization term is the L2-norm of the coefficient vector, which is the sum of the squared coefficients multiplied by the regularization parameter lambda. The regularization parameter lambda controls the strength of regularization applied to the coefficients. A higher lambda value results in stronger regularization.

Matrix notation:

$$\min_{\beta} \ \beta^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\beta - 2\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y} + \lambda\beta^{\mathrm{T}}\beta$$

The objective function can be represented in matrix notation using the design matrix X (containing the feature variables), the target vector y, and the coefficient vector beta. The matrix notation allows for a more compact and efficient representation of the objective function and facilitates optimization using linear algebra techniques. Intercept Term:

The regularization term does not affect the intercept term (beta_0) to ensure that the intercept is not penalized.

where:

$$\lambda = \text{lambda vector}$$
$$\lambda_1 = 0, \text{ the penalization doesn´t affect the intercept}$$
$$\lambda_{j\neq 1} = l_2 \text{ penalization (alpha)}$$

The provided approach formulates the Ridge regression problem as a convex optimization problem with a quadratic objective function and linear constraints. By leveraging the Gurobi quadratic programming solver, it efficiently finds the optimal coefficients that minimize the sum of squared errors while controlling for overfitting through regularization.

4.2 Elastic Net Procedure

Once again, in the experimental setup, the Elastic Net procedure was implemented utilizing both scikit-learn and Gurobi optimization software. In the scikit-learn implementation, an Elastic Net regression model was configured, integrating both L1 and L2 regularization techniques for comprehensive regularization. The model's hyperparameters, including alpha and l1_ratio, were fine-tuned through an exhaustive grid search, providing optimal values for regularization strength and the mixing parameter between L1 and L2 penalties. The best parameters ended up being: {'alpha': 0.0494, 'l1_ratio': 0.9}.

Transitioning to the Gurobi implementation, the Elastic Net procedure underwent a reformulation into a linear programming (LP) problem. The elastic net model uses both, L1 and L2, regularization techniques to perform the regularization/regression task. It does so by assigning a weight, $l1\_ratio$, on the L1 penalty and a weight $1-l1\_ratio$ on the L2 penalty. It also uses an α penalization value on penalties. Using n as the number of rows in the dataset and z as the number of features, the cost function of the elastic net model is as follows:[4]

The elastic net model uses both, L1 and L2, regularization techniques to perform the regularization/regression task. It does so by assigning a weight, $l1\_ratio$, on the L1 penalty and a weight $1 - l1\_ratio$ on the L2 penalty. It also uses an α penalization value on penalties. Using n as the number of rows in the dataset and z as the number of features, the cost function of the elastic net model is as follows:

$$\text{Elastic Net Cost Function} = \frac{1}{2n} \cdot \sum_{i=1}^{n} (y_i - \sum_{j=1}^{z} X_{ij} B_j)^2 + \alpha \cdot \left(l1\_ratio \cdot \sum_{j=1}^{z} |B_j| + \frac{(1 - l1\_ratio)}{2} \cdot \sum_{j=1}^{z} B_j^2 \right)$$

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

To perform the optimization task, I aim to find the beta coeifficients that minimize the value of the cost function above. The formulation of the optimization problem is as follows:

Variables:

β_i for i $\in n$: where beta represents the coeifficients of the features and the intercept
abs_β_i for i $\in n$: which is the absolute value of the beta coeifficients

Note: Since having $|\beta_i|$ would turn the optimization task into a quadratic programming problem, I had to implement the extra variable being the absolute value of the beta to ensure the problem remains and LP problem.

[4] Scikit-Learn. (2024). Sklearn.linear_model.ElasticNet. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Objective Function:

$$\text{MINIMIZE } \frac{1}{2n} \cdot \sum_{i=1}^{n} (y_i - \sum_{j=1}^{z} X_{ij} B_j)^2 + \alpha \cdot \left(l1\_ratio \cdot \sum_{j=1}^{z} abs\_\beta_i + \frac{(1 - l1\_ratio)}{2} \cdot \sum_{j=1}^{z} B_j^2 \right)$$

The values of α and $l1\_ratio$ will be the same values that were used in the above sklearn implementation of the elastic net model, thus the optimization function would run to minimize the objective function with respect to β

Constraints:

$$abs\_\beta_i \geq \beta_i \text{ for i} \in n$$
$$abs\_\beta_i \geq -\beta_i \text{ for i} \in n$$

By adding those two constraint, I have ensure that the value of abs_β_i is equal to $|\beta_i|$ while ensuring the problem is still an LP problem

The final step to all models experimental setups involved a detailed analysis, including not only performance metrics but also a meticulous examination of non-zero coefficients and the intercept, found in Section 5, Results. This evaluation highlighted the effectiveness of the Gurobi optimization approach in tackling the Elastic Net regularization problem as well as Lasso and Ridge, showcasing its accuracy in parallel with scikit-learn.

Section 5 : Results and Discussion

After running the 3 models with a black-box implementation as well as a loss function minimization, we were able to pull results from coefficient values (visuals), as well as a chart with SSE, R^2, Intercept and # of non 0 coefficients.

5.1 Coefficient Value Comparison

In the results section, we present the findings from the coefficient charts, comparing the coefficients obtained through the black-box approach and Gurobi optimization for three distinct models: Lasso, Ridge, and Elastic Net. The detailed results for each model are outlined in Appendix A (Lasso), Appendix B (Ridge) and Appendix C (Elastic Net). Despite observing similarities in the coefficients between the two methods (blackbox and gurobi), some patterns emerge. Notably, G2 and G1 prove to be the most influential categories, indicating their significant impact on the models' predictions. This is in line with what we were expecting, as grades from semester 1 and semester 2 would be the most telling for the final grades. Additionally, it is evident that Lasso tends to select less features compared to Ridge and Elastic Net. We will evaluate if this has a positive or negative impact on the accuracy when we look at the SSE and R^2 results.

However, by looking at the three charts individually, it is hard to see differences between models themselves. To do so, we present a chart exclusively for Gurobi coefficients, detailed in Appendix D. Noteworthy insights from this chart include the consistent prominence of G2 and G1, further affirming their significance across all models. Surprisingly, features such as father education, failures, alcohol consumption, extra-curricular activities, and romantic relationships exhibit negative coefficients. While some findings align with expectations, the negative impact of father education warrants further exploration. Despite minor variations, the three models generally provide comparable insights. Lasso, with fewer features, stands out, while Ridge and Elastic Net demonstrate closer similarities. We can also see that in almost all features, Ridge has the highest coefficients. We are not sure if this is good or bad, hence why the next section will discuss the evaluation metrics between all models.

The next section will dive into the evaluation metrics across all models, providing a comprehensive assessment of their performance and aiding in the determination of the most suitable model for the given dataset.

5.2 Evaluation Metric Comparison

After running the 3 models with a black-box implementation as well as a loss function minimization, the following results were found:

Chart 1: Evaluation Metric Results for all 3 Models

	Lasso		**Ridge**		**Elastic Net**	
	Blackbox	**Gurobi**	**Blackbox**	**Gurobi**	**Blackbox**	**Gurobi**
SSE	373.73	373.33	391.60	350.85	379.91	347.63
R^2	0.7692	0.7695	0.7582	0.7834	0.7655	0.7854

Intercept	10.33	10.32	10.33	10.33	10.33	10.23
# of non 0 coeffs	14	14	22	22	21	22

The evaluation of the Lasso, Ridge, and Elastic Net models, comparing the results obtained from the black-box approach and Gurobi optimization, reveals intriguing insights into their performance. In the context of the Sum of Squared Errors (SSE), we observe slight variations between the black-box and Gurobi results across all three models. Notably, the Elastic Net model demonstrates a notable reduction in SSE when optimized using Gurobi, suggesting that the optimization process has a positive impact on the model's predictive accuracy. Meanwhile, Lasso and Ridge models exhibit minimal changes in SSE, emphasizing the stability of their predictions between the two approaches.

When evaluating the coefficient of determination (R^2), the Gurobi-optimized models consistently outperform their black-box counterparts. The improvement in R^2 values across Lasso, Ridge, and Elastic Net signifies the enhanced explanatory power and goodness of fit achieved through the optimization process. The intercept values remain relatively consistent, reflecting the stability of the models' baseline predictions.

Additionally, examining the number of non-zero coefficients provides insights into feature selection. Both Lasso and Ridge models maintain a consistent count of non-zero coefficients between the black-box and Gurobi approaches, emphasizing the inherent sparsity enforced by Lasso and the continuous nature of Ridge. On the other hand, the Elastic Net model shows a slight increase in the number of non-zero coefficients when optimized using Gurobi, suggesting a nuanced balance between L1 and L2 regularization components.

If we decide our primary criteria for model selection are minimizing the Sum of Squared Errors (SSE) and achieving a high R-squared (R2) value, and considering that the Gurobi-optimized Elastic Net model demonstrates the lowest SSE and high R^2 values among the three models, it suggests that the Elastic Net model may be the most suitable for our specific dataset and objectives. Elastic Net combines both L1 and L2 regularization, providing a balance between feature selection (similar to Lasso) and maintaining the stability of continuous coefficients (similar to Ridge). The fact that the Gurobi-optimized Elastic Net outperforms its black-box counterpart in terms of SSE and R^2 further supports its effectiveness in improving predictive accuracy. It's important to consider other factors as well, such as the interpretability of the model, the significance of individual features, and the sparsity of coefficients. However, based on the provided information regarding SSE and R^2, the Gurobi-optimized Elastic Net appears to be a strong candidate for the "best" model in our context.

Section 6 : Challenges and Learning Outcomes

6.1 Project Challenges

Due to our access to many scholarly resources, we did not have too many challenges when writing the code or the report for this project. There are two small ones below, but mostly this project was well developed.

Challenge: Algorithmic Complexity

- Implementing the Gurobi optimization for Lasso Ridge and Elastic Net presented challenges due to the complex nature of the underlying quadratic programming problems. Gurobi requires formulations that transform non-linear components, such as the L1-norm, into linear expressions. Also, as seen in the results, we did not get the same coefficients for both the black box and the loss function.
- Solution: To address this, we reformulated the problem, introducing auxiliary variables and constraints to model the non-linear terms. This allowed us to leverage Gurobi's quadratic programming capabilities effectively. To the second point about coefficient similarity, we decided to view this as an insight instead of as a roadblock.

Challenge: Model Evaluation Consistency

- Ensuring consistency in evaluating the performance metrics across different implementations posed a challenge. The need for comparable metrics like SSE and R-squared required careful consideration of the evaluation process.
- Solution: We developed standardized evaluation procedures to calculate metrics consistently across both black box (Scikit) and Gurobi implementations. This involved close alignment of test sets and result interpretation.

6.2 What we have learned

Throughout this project, the whole team has learned a great deal on elastic net. We had already seen Ridge and Lasso loss functions in class, but it was interesting to see how these two functions were merged together in order to get a hybrid of the both, elastic net. Below are some points we thought were insightful during our research.

- Insights into Optimization Techniques: The project provided valuable insights into leveraging optimization techniques, specifically quadratic programming, for solving complex machine learning problems. Understanding the nuances of formulating optimization problems for algorithms like Lasso and Elastic Net deepened our understanding of underlying mathematical principles.
- Bridging Theory and Implementation: Implementing algorithms like Lasso Ridge and Elastic Net with Gurobi reinforced the importance of bridging theoretical knowledge with practical implementation. It necessitated translating mathematical formulations into functional code and dealing with real-world constraints.
- Model Comparison and Interpretation: Comparing results between black box and Gurobi implementations enhanced our ability to critically assess model performance. Understanding

differences in optimization outcomes, intercepts, and coefficients provided a different perspective on the strengths and limitations of each approach.

- Problem-solving and Adaptability: Encountering challenges in algorithmic complexity and consistent evaluation demanded problem-solving skills. The project encouraged adaptability, requiring us to refine formulations, troubleshoot, and implement solutions.
- Team Collaboration and Communication: Collaborating on this project emphasized effective team communication and coordination. Working on 1 code with multiple students at the same time can be tricky, but we found a schedule that worked for us.

In summary, the project not only enhanced our technical skills in implementing advanced machine learning algorithms but also encouraged problem-solving, collaboration, and practical application of optimization techniques. The challenges encountered served as valuable learning opportunities, contributing to a deeper understanding of both the theoretical and practical aspects of machine learning optimization and the concepts of elastic net.

Section 7 : Conclusion

7.1 Summary of key findings and results

In summary, the optimization through Gurobi contributes positively to the performance metrics of all three models, particularly evident in the improvement of R^2 values. The stability of SSE and non-zero coefficients across Lasso and Ridge highlights their robustness, while the Elastic Net model exhibits sensitivity to the optimization process. These findings provide valuable insights into the impact of optimization on model performance and guide further exploration of the interplay between regularization methods and optimization techniques. Based on the SSE value and the R^2, we have concluded that the gurobi model from elastic net has the best accuracy for our dataset. It was also interesting to see that the main insights from the data itself was that G2 and G1 are the most telling factors of success and failures, alcohol, extra-curriculars, and romantic relationships can actually negatively impact a students success (G3 grade). That might be important to keep in mind for the MMA program.

7.2 Suggestions for future improvements or extensions of the project

If we were to continue this research, there are many ways in which our project could be extended. Below is a list of potential ideas.

1) Exploration of Additional Regularization Techniques: Future work could involve the exploration and implementation of other advanced regularization techniques, such as group lasso or sparse group lasso, to further enhance the model's ability to handle complex datasets with correlated features.
2) Integration of Automated Hyperparameter Tuning: Implementing automated hyperparameter tuning techniques, like Bayesian optimization, could be considered. This would streamline the process of finding optimal hyperparameters, potentially improving model performance.
3) Incorporation of Feature Engineering: Extending the project to include more sophisticated feature engineering techniques may contribute to better capturing underlying patterns. Exploring interactions between features or incorporating domain-specific knowledge could enhance model interpretability.
4) Deployment and Integration: Integrating the developed models into real-world applications and deployment scenarios is a crucial next step. This involves considerations such as model interpretability, real-time predictions, and seamless integration with existing systems.
5) Evaluation on Diverse Datasets: Testing the models on diverse datasets from different domains could provide insights into their generalization capabilities. Understanding how the models perform on varied data distributions contributes to their robustness.

In conclusion, this project laid the foundation for implementing advanced machine learning algorithms using optimization techniques. The future extensions and improvements suggested aim to enhance the versatility, efficiency, and applicability of the models in diverse real-world scenarios.

Section 8 : References

Brownlee, J. (June 12th 2020). How to Develop Elastic Net Regression Models in Python. *Machine Learning Mastery.* https://machinelearningmastery.com/elastic-net-regression-in-python/

CFI Team. (2024). Elastic Net. *Corporate Finance Institute.* https://corporatefinanceinstitute.com/resources/data-science/elastic-net/

Dhumne, S. (March 12th 2023). Elastic Net Regression detailed guide. *Medium.* https://medium.com/@shruti.dhumne/elastic-net-regression-detailed-guide-99dce30b8e6e

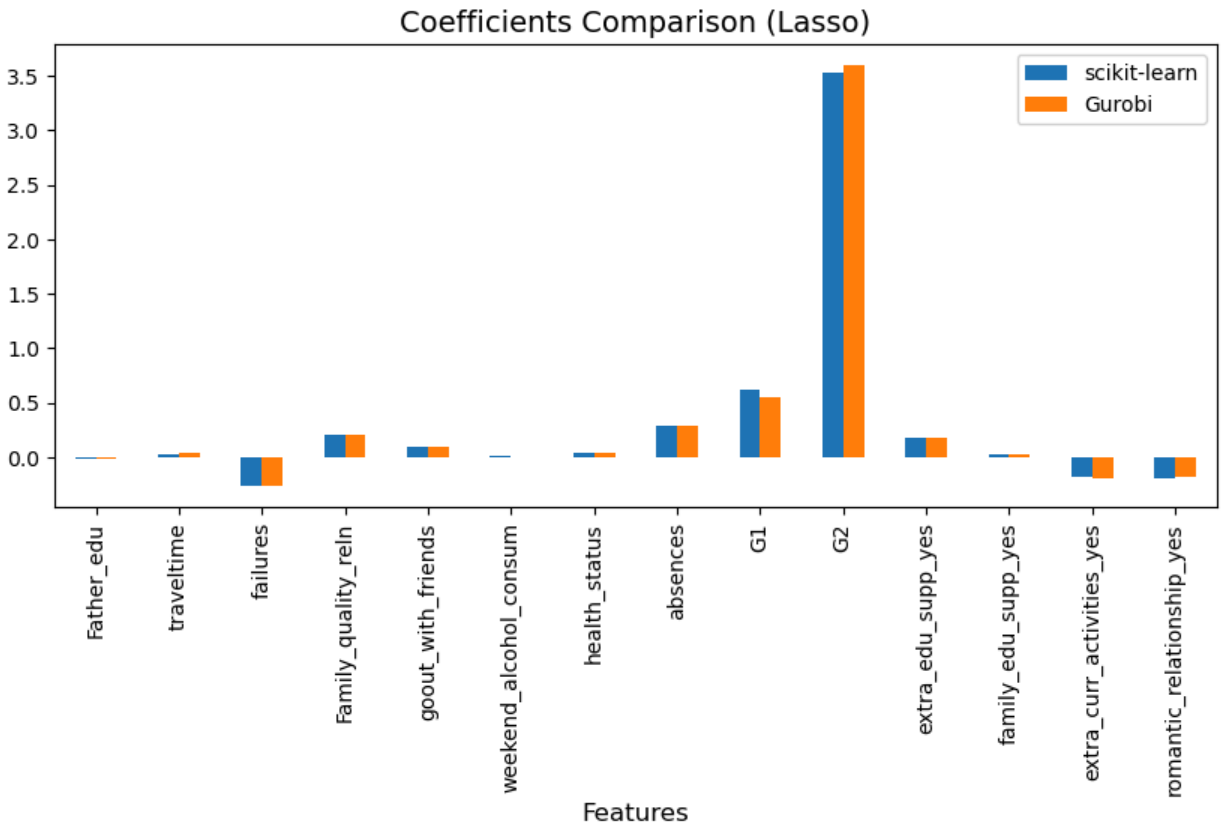
Giba, L. (2024). Elastic Net Regression Explained, Step by Step. *Machine Learning Compass.* https://machinelearningcompass.com/machine_learning_models/elastic_net_regression/

Kaggle. (January 2024). Student Final Grade Prediction. https://www.kaggle.com/datasets/tejas14/student-final-grade-prediction-multi-lin-reg

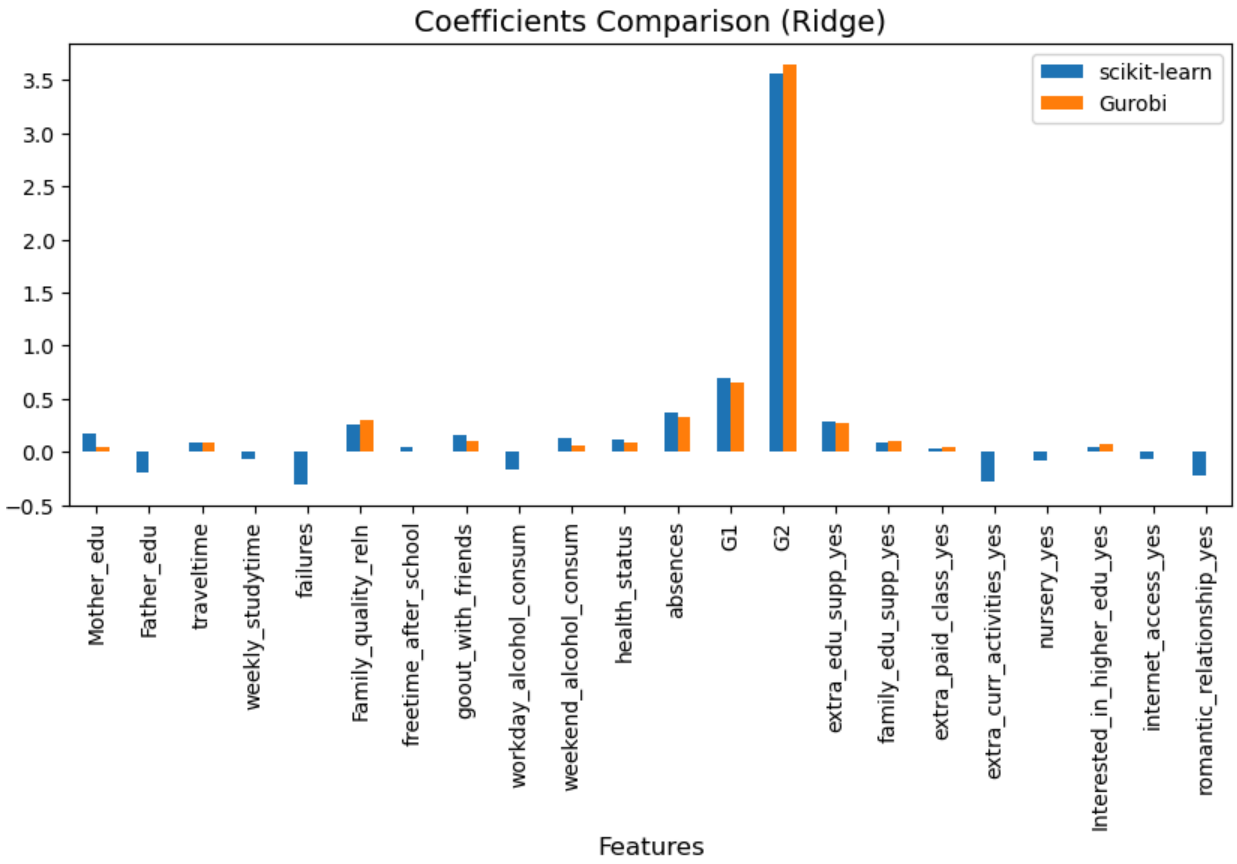
Scikit-Learn. (2024). Sklearn.linear_model.ElasticNet. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Section 9 : Appendices

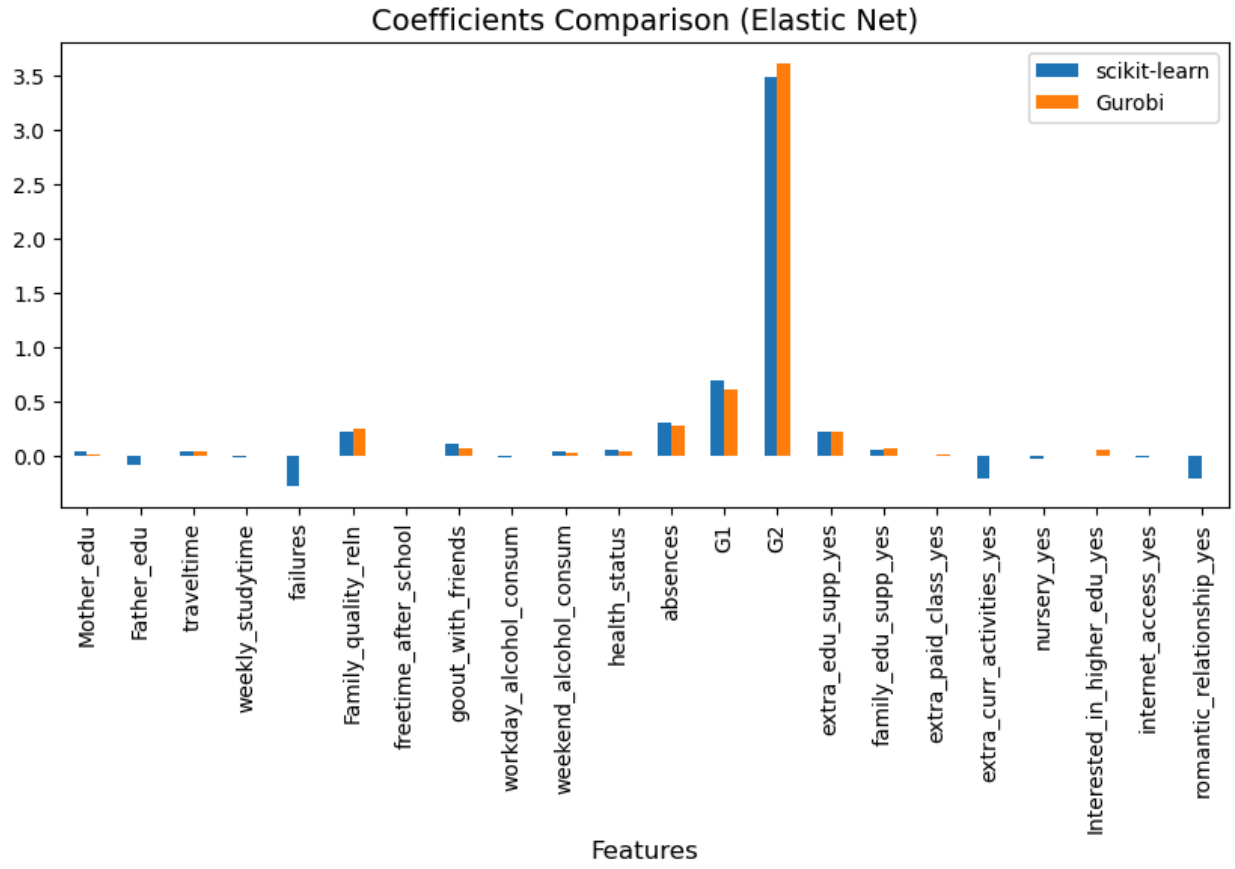
Appendix A: Lasso Coefficients Comparison Graph



Appendix B: Ridge Coefficients Comparison Graph



Appendix C: Elastic Net Coefficients Comparison Graph



Appendix D: Gurobi Coefficients Comparison for Lasso, Ridge and Elastic Net Graph