Performance Assessment D208 – Predictive Modeling  
Task II

Doug Haunsperger

College of IT, Western Governors University

December 20, 2023

# Part I. Rese**arch Questi**on

## A1. Question Proposal

I am proposing to examine the question, “Can LASSO regression both predict ancillary charges for a given patient at time of admission, as well as identify the factors that contribute the most to these charges?”. As a business, the hospital has an interest in predicting future revenue from a new patient. For public health reasons, if there are controllable factors that lead to higher charges (i.e. more severe interventions are required), a public information campaign could be conducted to persuade people to change their behavior.

## A2. Goal

I will determine if any of the other data collected in our dataset, that would be available at time of admission (i.e. demographic and medical history), have a correlation and possible causal relationship with ancillary charges (‘Additional\_charges’) made by the hospital for miscellaneous procedures and treatment not related to the length of stay in the hospital. By using LASSO regression (“Least Absolute Shrinkage and Selection Operator”) I am attempting to focus on the most important data features from the wide variety of demographic and medical history variables available.

# Part II. Method Justification

## B1. Explanation of Prediction Method

LASSO regression is used “to find a balance between model simplicity and accuracy” and “simple, sparse models (i.e. models with fewer parameters)” (Kumar, 2023). It can be used to “fit a model containing all possible predictors and … perform variable selection by using a technique that … shrinks the coefficient estimates towards zero.” (Kirenz, 2021).

In Ordinary Least Squares multiple regression, the objective is to minimize the residual sum of squares, given by Kirenz as:

LASSO adds an additional term to the minimization function, again per Kirenz:

In this equation, is a hyperparameter that controls the amount of penalty given to large coefficients (), otherwise known as model parameters (Kumar, 2023).

I will use *k*-fold cross-validation to find an optimal value for . The expected outcome is to have a model that performs as well or better than an OLS multiple regression model, but with fewer parameters.

## B2. Method Assumptions

LASSO, as a linear regression model, shares the underlying assumptions of any linear model, namely per Liu et al. (2016):

1. Linearity – there exists a linear relationship between each predictor variable and the response variable.
2. Homoscedasticity – the residual variances are the same for any value of the predictor variables.
3. Independence – each observation in the dataset is unrelated to the others.
4. Normality – for any fixed set of predictor variables, the response variable output is normally distributed.

In addition, to perform feature selection, LASSO assumes sparsity – that some independent variables are irrelevant to the target – and the irrepresentable condition – that relevant variables have minimal correlation with irrelevant variables. (Vasconcelos et al., 2017).

## B3. Packages List

Figure 1  
*Package and library import statements*

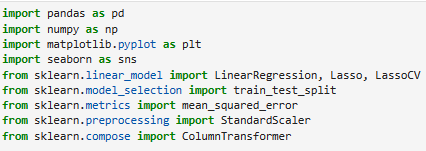


Figure 1 shows my import code. Each package or library imported is useful in the full scope of the project. Pandas is used to work with DataFrames. Numpy is used for various mathematical functions (finding means, natural logarithms, etc.). Pyplot and seaborn are used for graphical display.

The remaining lines import certain functions or classes from the scikit-learn package. From the API reference (2023), we see that LinearRegression and Lasso are used to perform the OLS and LASSO linear regressions. LassoCV is used for cross-validation to tune the hyperparameter . Train\_test\_split automates the splitting of a dataset into training and testing data. Mean\_squared\_error is a scoring function that calculates the mean squared error (MSE) metric for the model. I used StandardScaler to scale the quantitative variables to unit variance with zero mean. Finally, the ColumnTransformer function allowed me to easily scale & transform the quantitative variables while leaving the categorical columns untouched.

# Part III. Data Preparation

## C1. Data Preprocessing

To clean my data in preparation for the linear regression analysis, I checked for and removed outliers in the given data.

Code and explanation of methods is in the attached Jupyter notebook, “LassoReg.ipynb” – see section C1.

## C2. Data Set Variables

Variable selection and classification are shown in the attached Jupyter notebook, section C2 (appears in the code above section C1). Note that only variables that would be available at the time of hospital admission are selected in order to provide a *predictive* model.

## C3. Data Preparation & Transformation

I performed several steps to further transform the data prior to running the regression analysis. First, I removed the outliers identified in section C1. I needed to transform all categorical values into numeric data. Yes/No values can easily be encoded as 1/0. Complication\_risk is ordinal and can be encoded as 1/2/3 for Low/Medium/High. Gender, Marital, and Initial\_admin contain nominal data, not ordinal, so need to be transformed to a one-hot encoding. Finally, I performed a log transformation on Population and Income to make their distributions closer to normal.

Code and references are given in the attached Jupyter notebook, section C3.

## C4. Cleaned Data Set

The cleaned and transformed data set used for the regression analysis is attached as ‘lasso\_clean\_medical\_data.csv’.

# Part IV. Model Comparison & Analysis

## D1. Train/Test Split

See attached Jupyter notebook, section D1. I used the train\_test\_split function to split the cleaned data set into an 80/20 train/test split. The files are output to CSV at the end of the notebook and are attached to the submission as ‘lasso\_train\_medical\_data.csv’ and ‘lasso\_test\_medical\_data.csv’.

## D2/D3. Analysis Technique, Intermediate Calculations, Code Output

See attached Jupyter notebook, section D2, for code and output.

“Lasso performs best when all numerical features are centered around 0 and have variance in the same order.” (Kirenz, 2021). I started the analysis by scaling the quantitative columns using the StandardScaler function.

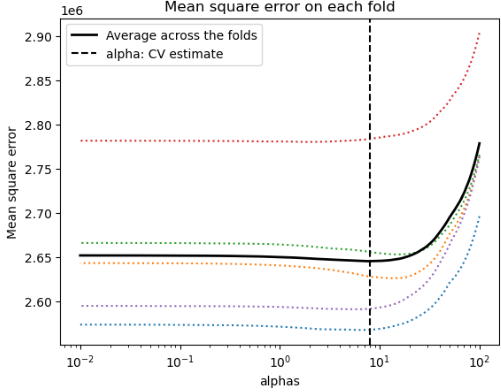
Next, I instantiated a LASSO model with an arbitrary hyperparameter, as well as an OLS multiple regression model for comparison. The mean squared error and R2 values for these initial models on the test data were as follows:

|  |  |  |
| --- | --- | --- |
|  | OLS | LASSO () |
| R2 | 0.93849 | 0.93852 |
| MSE | 2.675 ∙ 106 | 2.675 ∙ 106 |

I then used 5-fold cross-validation (via the LassoCV function) to tune the hyperparameter. Note in the video submission, you will see a slightly different calculated (8.09 vs 8.067), since initially I used a linear space across potential alphas between 0.001 and 100. I decided that a logarithmic space made more sense after recording the video.

Finally, I plotted a curve of mean squared error vs. the different alphas for each fold of the cross-validation, as well as the mean across all folds. That curve is reproduced in Figure 2.

Figure 2  
*MSE vs LASSO alpha hyperparameter*



# Part V. Data Summary & Implications

## E1. Accuracy and MSE

Using the best hyperparameter found by cross-validation, the final R2 and MSE of the LASSO model on the test data compared to a base OLS looks like this:

|  |  |  |
| --- | --- | --- |
|  | OLS | LASSO () |
| R2 | 0.93849 | 0.93856 |
| MSE | 2.675 ∙ 106 | 2.673 ∙ 106 |

These R2 values show an excellent fit for both the original OLS and the trained LASSO model. The mean squared error is slightly (~0.1%) better in the LASSO case.

The mean value for Additional\_charges across the full data set was $12,946. The root mean squared error for the trained LASSO model is $1635 giving a reasonable expectation that the predicted value on admission will be close to the final value.

## E2. Results & Implications

Both the OLS model and the LASSO model have good results when looking at R2 and MSE. The benefit of the LASSO model is in the feature reduction. The OLS model has 26 features with non-zero coefficients, 12 of which affected the predicted output by at least $50 per unit change in the feature. By contrast, the LASSO model only has 14 features with non-zero coefficients, of which only 8 affect the output by at least $50 per unit change. This implies that the organization should focus on these 8 elements when predicting additional patient charges.

## E3. Limitation

This analysis does not prove that the data set satisfies the irrepresentable condition that is an underlying assumption for the validity of LASSO. Jia & Rohe (2012) give a method for proving this, but it is beyond the scope of this project. It is possible that if the dataset violates the irrepresentable condition, then an irrelevant feature could be erroneously selected (Vasconcelos et al., 2017).

## E4. Recommendations

I recommend the hospital adopt the simpler LASSO model to predict future revenue. In addition, as mentioned in section A1, if a controllable factor is found that highly influences the cost of hospital stays, a public information campaign can be produced to recommend that people better control this factor. In this case, high blood pressure has a dramatic impact on the expected hospital cost, on average adding over $8500 to the bill. I recommend this information be shared with primary care providers to help them convince their patients to maintain a healthy blood pressure prior to landing in the hospital.

# Part VI. Demonstration & Supporting Documentation

## F. Demonstration Video

A video describing my methods and code can be found at: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=fe65a348-f7f3-41a4-bbae-b0de00fbaa12>

## G. Third-party Code Sources

Bengtsson, P. (2013, June 14). *In Python you sort with a tuple*. Peterbe.com. <https://www.peterbe.com/plog/in-python-you-sort-with-a-tuple>

Kassies, R. (2013, May 6). *Make more than one chart in same IPython Notebook cell.* StackOverflow. <https://stackoverflow.com/questions/16392921/make-more-than-one-chart-in-same-ipython-notebook-cell>

Larose, C., & Larose, D. (2019). *Data Science Using Python and R.* Wiley.

*scikit-learn 1.3.2 API reference documentation (2023).* [*https://scikit-learn.org/stable/modules/classes.html#*](https://scikit-learn.org/stable/modules/classes.html)

*scikit-learn 1.3.2 User Guide* (2023). [https://scikit-learn.org/stable/modules/compose.html#](https://scikit-learn.org/stable/modules/compose.html)

## H. References

Jia, J. & Rohe, K. (August 28, 2012). *Preconditioning to comply with the Irrepresentable Condition.* arXiv. <https://arxiv.org/pdf/1208.5584.pdf>

Kirenz, J. (December 27, 2021). *Lasso Regression with Python*. Kirenz.com. <https://www.kirenz.com/post/2019-08-12-python-lasso-regression-auto/>

Kumar, D. (May 30, 2023). *A Complete understanding of LASSO Regression*. Great Learning. <https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/>

Liu, C., Milton, J., & McIntosh, A. (January 6, 2016). Simple Linear Regression. *Correlation and Regression with R.* Boston University School of Public Health. <https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html>

Middleton, K. (November 2022). *Getting Started with D208, Part I.* Western Governors University. <https://westerngovernorsuniversity.sharepoint.com/:b:/r/sites/DataScienceTeam/Shared%20Documents/Graduate%20Team/D208/Student%20Facing%20Resources/Dr.%20Middleton%20Getting%20Started%20with%20D208(Part%20I)COIT.pdf?csf=1&web=1&e=CLRFMI>

Vasconcelos, G., Hoeltgebaum, H., Fonseca, Y., & Milagres, T. (June 14, 2017). *When the LASSO fails???* InsightR blog. <https://insightr.wordpress.com/2017/06/14/when-the-lasso-fails/>