Economics 4803/8803: Machine Learning for Economics Problem Set 2

Due by: **March 1, 5 PM ET**

This problem set is based on topics including Supervised Learning: Linear models. You will use Airbnb dataset for New York City for this and the subsequent problem sets. You can learn more about this dataset, along with information on select variables here. Here is the detailed data dictionary.[1]

You are allowed to work in groups of up to four students, but you must disclose the members of your groups. Individual submissions are required. The code you submit may be identical to the one of the other group members, but I expect comments and answers to the questions to be your own.

Your submission should consist of:

1. a pdf file with responses to the questions marked with * (please do not attach any code snippets)
2. a Rdocument with code
3. you should upload these materials via the course's Canvas website (please do not email me with your home-work submission).

1 Empirical Problems

This part of the problem set is designed to perform predictive tasks using linear models. The task of this problem set is to predict the price of an Airbnb listing using listing characteristics.

1. Preparing data for the analysis (you don't have to report the results of any of these in the pdf):

[1]Note that Airbnb updates this data every few months. For the purpose of this course, we will use the **June 5, 2023** version of this data.

(a) Download the data airbnb_data.csvfrom Canvas and bring it to R.[2]

(b) Remove observations with missing or NA values in price. This is the outcome variable.

(c) Remove observations with or NA values in accommodates, beds, number_of_reviews, and review_scores_rating.

(d) Create a variable called host_experience which is the number of days be- tween June 5, 2023 and host_since. Divide this by 365 to convert this in number of years. Remove observations with missing values. This is a rough proxy for host experience in years. Remove any NA values.

(e) Create a variable called entire_apt. Assign a value of 1 if the variable room_type is "Entire home/apt", otherwise 0. Remove any NA values.

(f) Create a variable host_is_superhost which takes a value 1 if a host is a su- perhost and 0 otherwise. Use the following algorithm: a host is a superhost if their response rate (host_response_rate) is greater than or equal to 90% and total number of reviews (number_of_reviews) is at least 10 and their review rating (review_scores_rating) is at least 4.8.[3] Remove any NA values.

(g) Sort your data by id.

2. Analysis:

(a) Set the seed to 0 (set.seed(0)) so that your results are comparable to the solutions.

(b) Begin by randomly allocating 50% of the data to the test sample and the rest to the training sample.

(c) Estimate a linear regression model with accommodates, beds, host_experience, host_is_superhost, entire_apt, number_of_reviews and review_scores_ratingas covariates and priceas the outcome variable. Compute the R squared and calculate the MSE of the test sample. *

rsquared_linear	0.157881570368175
mse_linear	19217.3448696689

(d) Estimate a linear model with second order polynomials for accommodates, beds, host_experience, number_of_reviews, review_scores_rating, all in-

[2]You can download the data yourself from this link: http://insideairbnb.com/get-the-data. Download the file listings.csv.gz. Make sure that the dataset version is June 5, 2023. When you extract the gzip archive, it should be a 107.6 MB csv file called listings 2.csv. However, make sure to run build_airbnb_data.Rbefore attempting the problem set.

[3]You may use ifelse()command for this purpose.

teractions/cross terms of the above five variables, and host_is_superhost, entire_apt (so, total of 22 terms). Compute the R squared and calculate the MSE of the test sample.*

rsquared_poly	0.17883759889052

mse_poly	18988.4540967789

(e) Perform a simple backward stepwise selection procedure on the model in (d) using Algorithm 6.3 from ISL (page 231) using BIC and R squared as criteria. Compute the out-of-sample MSE of the selected models. Compare with the fit and prediction performance of the OLS.*

mse_step	19051.0005873782

The initial linear model provided a baseline with an MSE of 19217.34. Incorporating polynomial features slightly improved the model fit, reducing the MSE to 18988.45, albeit at the cost of increased complexity. The backward stepwise selection aimed to optimize this complexity by selectively removing less informative variables, resulting in a marginally adjusted MSE of 19051.00. This stepwise refinement, while simplifying the model, did not significantly outperform the original linear model in terms of predictive accuracy. This suggests that the added complexity of polynomial features and their interactions offers limited benefits for prediction accuracy in this context, highlighting the importance of balancing model complexity against practical performance gains.

(f) For the model in (d), fit both a Ridge regression and a Lasso, each for three values of the tuning parameter: 0, 5, and 10. Create a table that summarizes your estimates for both the models and for each levels of tuning parameters. What happens as the penalty parameter increases? Compute the out-of-sample MSE for all the models and comment on the predictive performance.*

mse_ridge	Named num [1:3] 19217 19226 19243

mse_lasso	Named num [1:3] 19217 19323 19540

As the penalty parameter increases in both Lasso and Ridge regression models, the Mean Squared Error (MSE) for out-of-sample predictions slightly rises. For Lasso, the MSE grows from 19217 with no penalty to 19540 with a penalty of 10, and for Ridge, it increases from 19217 to 19243 over the same range of penalties. This trend suggests that higher regularization leads to a modest decrease in predictive performance for this dataset, likely because the models become overly simplistic and potentially underfit the data. The initial model without a penalty appears to be sufficiently complex to capture the underlying patterns without significant overfitting, indicating that additional

regularization does not improve, and may slightly hinder, the model's accuracy on new, unseen data. This emphasizes the importance of fine-tuning the penalty parameter to maintain a balance between reducing overfitting and preserving the model's predictive capabilities.

(g) Now, for both Lasso and Ridge, perform a 10-fold cross validation procedure to pick the tuning parameter, using the training dataset only. Compute the MSE in the test sample for both Lasso and Ridge. Comment on predictive performance for both models, and compare it with the models in (c)-(f).*

mse_cv_lasso	19217.2520818975
mse_cv_ridge	19228.1513233035

After performing a 10-fold cross-validation procedure to optimize the tuning parameters for both Lasso and Ridge regression models, the Mean Squared Error (MSE) for the test sample is 19217.25 for Lasso and 19228.15 for Ridge. These results provide an interesting comparison to the models evaluated earlier in sections (c)-(f).

The MSE for both Lasso and Ridge after cross-validation is very close to the MSEs observed for the initial linear model and the models with manually selected penalty parameters. Specifically, the cross-validated Lasso model shows a slight improvement over the manually tuned models, suggesting that the optimized penalty parameter effectively balances bias and variance to enhance prediction accuracy marginally. The cross-validated Ridge model also demonstrates a similar trend, though its MSE is slightly higher than that of the Lasso model, indicating a minor difference in how each regularization technique impacts model performance.

Comparing these results with the earlier models, it's evident that cross-validation provides a more systematic and data-driven method for selecting the penalty parameter, potentially leading to slight improvements in predictive performance. However, the differences in MSE are relatively minor, suggesting that for this dataset, the benefits of regularization and the specific choice of the penalty parameter are somewhat limited. This could be due to the nature of the data or the variables selected for the model, where either overfitting is not a significant issue, or the models are already capturing the essential patterns in the data with the given variables.

In summary, the cross-validation approach for optimizing the penalty parameters in Lasso and Ridge regression slightly refines the models' predictive accuracy. Still, it does not lead to substantial improvements over the basic linear regression model or the models with fixed penalty parameters. This underscores the importance of understanding the dataset and the context in which the models are applied, as the benefits of regularization and parameter tuning can vary depending on these factors.

(h) Now step back, and consider the prediction task of determining the price of the listing. What linear supervised learning procedure seems most appropriate given the sample size and covariate structure? Why do you think is that?*

Considering the task of predicting the price of an Airbnb listing, a simple linear regression model appears to be the most appropriate linear supervised learning procedure given the sample size and covariate structure. This conclusion is drawn from the observation that both Lasso and Ridge regression, despite their regularization benefits, did not substantially improve the prediction accuracy over the basic linear regression model as evidenced by the MSE values obtained. The slight improvements offered by optimized penalty parameters through cross-validation indicate that overfitting is not a significant issue with this dataset. Moreover, the nature of the covariates—such as the number of accommodations, beds, host experience, and review scores—suggests a linear relationship with the price that does not overly benefit from the complexity reduction offered by Lasso or the shrinkage effect of Ridge regression. The simplicity of the linear regression model, its interpretability, and the absence of overfitting given the dataset characteristics make it an efficient choice for capturing the essential patterns in the data without the need for the additional complexity of regularization methods.

3. Repeat 2 (a) to (g) with the following variations:

(a) Add in three additional predictors that are highly correlated with host_experience, host_is_superhost, and number_of_reviews. Here is some example code:

datam$noise1 ← datam$host_experience + rnorm(nrow(datam), .01) datam$noise2 ← datam$host_is_superhost + rnorm(nrow(datam), .01) datam$noise3 ← datam$number_of_reviews + rnorm(nrow(datam), .01)

(b) Decrease the size of the training dataset by allocating 90% of the data to the test sample. Then allocate 98% of the data to the test sample.
Note: with three training sample sizes (50%, 10%, and 2%) and two sets of covariates (one with noise variables and one without), there should be six sets of results. **Tip:** make this simple by writing your code in such a way that allows you to specify at the top which iteration you'd like to run with a change in sample size and/or a binary variable that specifies whether or not you would like to add noise. One way could be to write a function(s) and/or loops to implement these regressions

(c) Write a short note explaining why you think the MSE changes in the way that it does.

The variation in MSE across different models highlights several key points about model performance and data characteristics. When the training sample size decreases from 50% to 10%, and further to 2%,

there's a general trend of increasing MSE, especially noticeable in models without noise variables. This increment in MSE can be attributed to the reduction in training data, which limits the model's ability to learn the underlying patterns effectively, leading to poorer generalization to the test data.

Introducing noise variables slightly complicates this picture. While in some cases, such as with a 50% training sample, the introduction of noise does not significantly deteriorate the model's performance, in smaller training samples, the impact of noise becomes more pronounced, leading to higher MSEs. This suggests that as the amount of informative data decreases, the model becomes more sensitive to the inclusion of irrelevant predictors, which can obscure the true signal and degrade predictive accuracy.

The presence of highly correlated noise variables with key predictors like host_experience, host_is_superhost, and number_of_reviews adds redundancy and potential multicollinearity, complicating the model's ability to distinguish between relevant and irrelevant information. This effect is exacerbated in smaller training datasets, where the limited data points make it more challenging to accurately estimate the model parameters, leading to an inflated MSE.

Overall, the changes in MSE reflect the delicate balance between having enough data to train robust models and the impact of including noise or irrelevant variables, particularly in smaller datasets. The results underscore the importance of careful variable selection and the potential benefits of regularization techniques, like Lasso, that can help mitigate the effects of noise by penalizing the inclusion of less relevant predictors.

2 Conceptual Problems

All problems found in the textbook *Introduction to Statistical Learning* 2nd edition. Please keep your answers concise (2-3 lines).*

1. Complete Exercise 4 in Section 3.7 (page 122)

   (a) If the true relationship between X and Y is linear, the RSS for the linear regression model on the training data is expected to be similar to or slightly lower than that of the cubic regression model. This is because the linear model directly corresponds to the true relationship, minimizing the residuals. The cubic model, which includes X^2 and X^3 terms, could potentially fit the training data even more closely (thus possibly having a slightly lower RSS) due to its additional complexity, but this could also lead to overfitting, capturing noise in the training data as signal.

   (b) When using the test data, the linear regression model's RSS is expected to be lower or similar to that of the cubic regression model. This is because the simpler model, which correctly matches the true linear relationship, is less likely to overfit than the cubic model and therefore performs better or equally well on unseen data. The cubic model's extra terms may capture noise in the training data, leading to a higher RSS on the test data due to poor generalization.

   (c) If the true relationship between X and Y is non-linear and we're unaware of its deviation from linearity, it's challenging to predict which model would have a lower training RSS without more information. The cubic regression might better capture the true relationship due to its flexibility in modeling non-linear patterns, potentially resulting in a lower RSS compared to the linear model. However, the degree and nature of the non-linearity would significantly impact which model fits better; without specifics, it's difficult to definitively say which would perform better on the training data.

   (d) For the test RSS, if the relationship is non-linear, the cubic regression model could potentially perform better (lower RSS) than the linear model, especially if the cubic terms help in capturing the true nature of the relationship between X and Y. However, the performance of the cubic model on test data heavily depends on whether its complexity appropriately captures the underlying pattern without overfitting to the training data. If the cubic model overfits, its test RSS might be higher despite a lower training RSS. Conversely, if the non-linearity is mild or the cubic model accurately captures the underlying pattern without overfitting, it could indeed yield a lower test RSS compared to the linear model.

2. Exercises 2 and 3 of Section 6.6 (pages 282-284)

Exercise 2:

(a) iii is true because the method imposes a penalty on the size of coefficients, which can lead to some coefficients being shrunk to zero. This reduction in flexibility can increase bias because the model is constrained and may not fit the training data as closely as a least squares regression. However, by reducing the model's complexity, Lasso tends to decrease the variance of the predictions, which can lead to improved prediction accuracy if the increase in bias is offset by a larger decrease in variance. The key aspect of Lasso is its ability to perform variable selection, which can be particularly beneficial in scenarios with many predictors, some of which may be irrelevant to the prediction task.

(b) iii is true because the method imposes a penalty on the size of the coefficients. However, unlike Lasso, Ridge regression does not set coefficients to zero but instead shrinks them towards zero. This constraint increases the model's bias but decreases its variance. Ridge regression is particularly effective when there is multicollinearity among the predictors or when the number of predictors exceeds the number of observations. The improvement in prediction accuracy with Ridge regression over least squares comes from a trade-off where the increase in bias is smaller than the decrease in variance.

(c) i is true likely because it can model complex relationships that are not possible to capture with a straight line or a plane. This increased flexibility can lead to a lower bias since the model can fit the training data more closely. However, the increased flexibility can also lead to higher variance, as the model might become too tailored to the training data, capturing noise as if it were a real pattern (overfitting). The key to improved prediction accuracy with non-linear methods lies in their ability to capture the true underlying patterns in the data without overfitting, which requires careful tuning of model parameters and possibly regularization techniques to manage the trade-off between bias and variance.

Exercise 3:

(a) iv is true because with less regularization, the coefficients can grow larger to minimize the residuals, thereby decreasing the training RSS.

(b) ii is true because when s is too small, the model might be too simple and underfit the test data, leading to higher RSS. As s increases, the model fits better, reducing the test RSS up to a point. Beyond that, increasing s further can lead to overfitting, which would increase test RSS.

(c) iv is true because less regularization means more sensitivity to the training data, which increases variance as the model will capture more noise.

(d) ii is true because at a very low s, the strong constraint on the coefficients causes high bias because the model is too simple to capture the true relationship. As s increases, the model can fit the data better, decreasing bias. However, at high levels of s, the model starts to fit noise, not just the underlying relationship, which can increase bias again due to overfitting.

(e) v is true because the irreducible error is due to the variability in the data that cannot be explained by the model regardless of the choice of s.