MATH60049 Coursework 1

March 1, 2023

Linear Regression on the Diamonds dataset

Introduction

The dataset I will be using is a dataset listing properties of approximately 54 000 diamonds. I obtained this dataset from the Kaggle dataset repository. My motivation for choosing this dataset lies within its interesting relationships, including multiple colinearity between predictors, and interesting non-linear relationships. The dataset has 10 features, including an index variable X, categorical variables Cut, Color and Clarity, and numerical variables Carat, Depth, Table and dimensions X, Y and Z. The dataset can be used to predict either carat or price, but in this report we will initially predict price as a linear function of the other parameters. I find this to be the most interesting option, as the price of a gemstone is something seemingly abstract depending on its perceived value and beauty, and so the ability to predict this variable based on measurable qualitative and quantitative aspects will expose whether there truly is an underlying value.

Data exploration and preparation

We begin by transforming the ordinal categorical variables Cut, Color and Clarity into factors so that we can use them as linear regression predictors. We plot box diagrams of some of the variables (FIGURE 1) and see that there are outliers, so we remove any datapoints with features lying outside 1.5 interquartile ranges of the mean.

We then explore the relationships between each predictor variable and the variable Price, in Figure 1.

We see significant association between Price and most of the variables (excluding Table), suggesting a linear regression model may provide some predictive insight. We notice, however, that as the price and dimensions of the diamond increase, there is a much less clear relationship - the variance is much higher with the more expensive diamonds. We will explore this more later.

Ordinary Least Squares Regression

We will divide our data into training and test in an 80:20 split. We build a basic linear model trained on the training set:

```
lm(formula = price ~ ., data = data_main)
```

and achieve an MSE of 924354 on the test set. All variables are significant at $\alpha = 0.001$ except for Table, so we exclude this from the following linear model. We plot the standardised residuals versus leverage (see Figure 2) to identify influential observations, finding 2 influential points and 1 with an extremely high residual. We remove these from our dataset on the assumption that they are incorrect.

We will now improve our model by including transformations and interaction terms. We first suspect that volume will be a predictor of price, so we estimate volume by $X \times Y \times Z$. We also take the log of Carat, which exhibits right skedacity.

```
lm(formula = price~. - table, data=data_main)
```

Our MSE is slightly reduced to 909383. This is likely because there is a relationship between our crude estimation of the volume of the diamond, and the price. Intuitively this makes a lot of sense: people will pay more for bigger gemstones. Additionally, the Carat variable shows an exponential relationship with several others, so taking a logarithm is likely to help.

LASSO Regression

We now train a LASSO model on the training set. We will use the elastic net package in R, setting $\alpha = 1$, with repeated 10-fold cross validation to optimise λ. We call

```
model2 <- cv.glmnet(x_train, y_train, alpha=1, nlambda=20)
```

and find our optimal l_1 penalty to be 0.33543.

See Figure 4 for the out-of-sample cross-validation MSE for each scanned value of λ when optimising the model.

We use this to make predictions for the test set:

```
bestlam = model2$lambda.min
glm2_best <- glmnet(x_train, y_train, alpha=1, lambda=bestlam)
```

Our MSE is slightly reduced compared to the standard OLS model, down to 916342, however it is still not as good as our model with transformations and interactions.

We include these terms and re-train the model. This time, we find the optimal λ to be 0.17234, which is slightly lower than before indicating that our transformations reduce some of the variance of the linear model. Our MSE is now 906546, which is marginally better than the original model. However, such a difference is so negligible that we can conclude our original model is reasonably low-variance, and the l_1 penalty doesn't contribute much.

Ridge regression

We will now optimise a Ridge regression model on the dataset, this time using the Elastic Net function with $\alpha = 0$. We again use repeated 10-fold cross-validation to find the λ which minimises the MSE:

```
model3 <- cv.glmnet(x_train, y_train, alpha=0, nlambda=20)
```

and find our optimal l_2 penalty to be 1.831. We calculate our out-of-sample MSE:

```
bestlam3 = model3$lambda.min
glm3_best <- glmnet(x_train, y_train, alpha=0.5, lambda=0.05)
```

and find it is in fact higher that the previous model: 931493. Why is this? It is likely that the l_2 penalty overshrinks the coefficients on our trained model, when in fact, given the large amount of data it is trained on and the relationships present between the predictors and outcome Price, our optimal Ridge model actuall becomes too biased, which contributes greater to the MSE than the reduction in variance we achieve.

Transforming the dependent variable

It is clear we have been ignoring something when carrying out these analyses. Our dependent variable, Price, exhibits right-skewedness.

In addition to this, from Figure 2 we can see that several of the numerical variables have a non-linear, somewhat exponential relationship with Price.

A potential solution is to predict log(Price) with a linear model. This may yield much better estimates.

However, this will mean we cannot compare MSEs, as the MSE of the logarithm will be much smaller. Instead, we take the exponent of our predictions $\log(y_i)$ and use this to calculate the MSE.

We retrain a linear model on this test set, but instead use the logarithm of Price as the dependent variable:

```
model4 <- lm(log(price)~., data=data_main)
ypred4 = predict(model4, data_test)
MSE(exp(ypred4), y_test)
```

Our MSE has been shrunk to almost a third of what it was! We now have an MSE of 354726. This is the best model by far.

What does this tell us about the data? That perhaps a linear model is not the best choice. Further exploration, for example nonlinear models such as k-nearest neighbours and Random Forest regression, are likely to also yield better results than Linear Regression and shrinkage methods.

Appendix

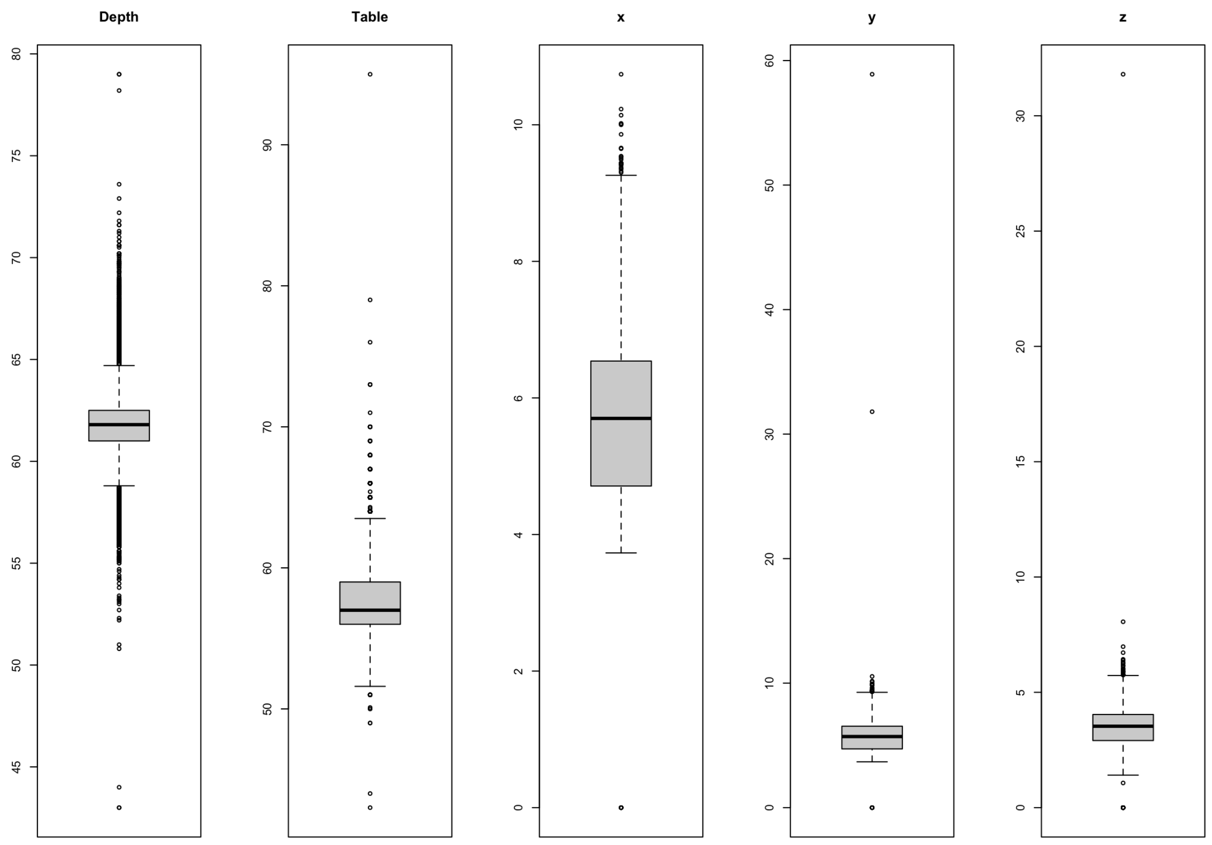

Figure 1: Boxplot of select predictors, showing outliers

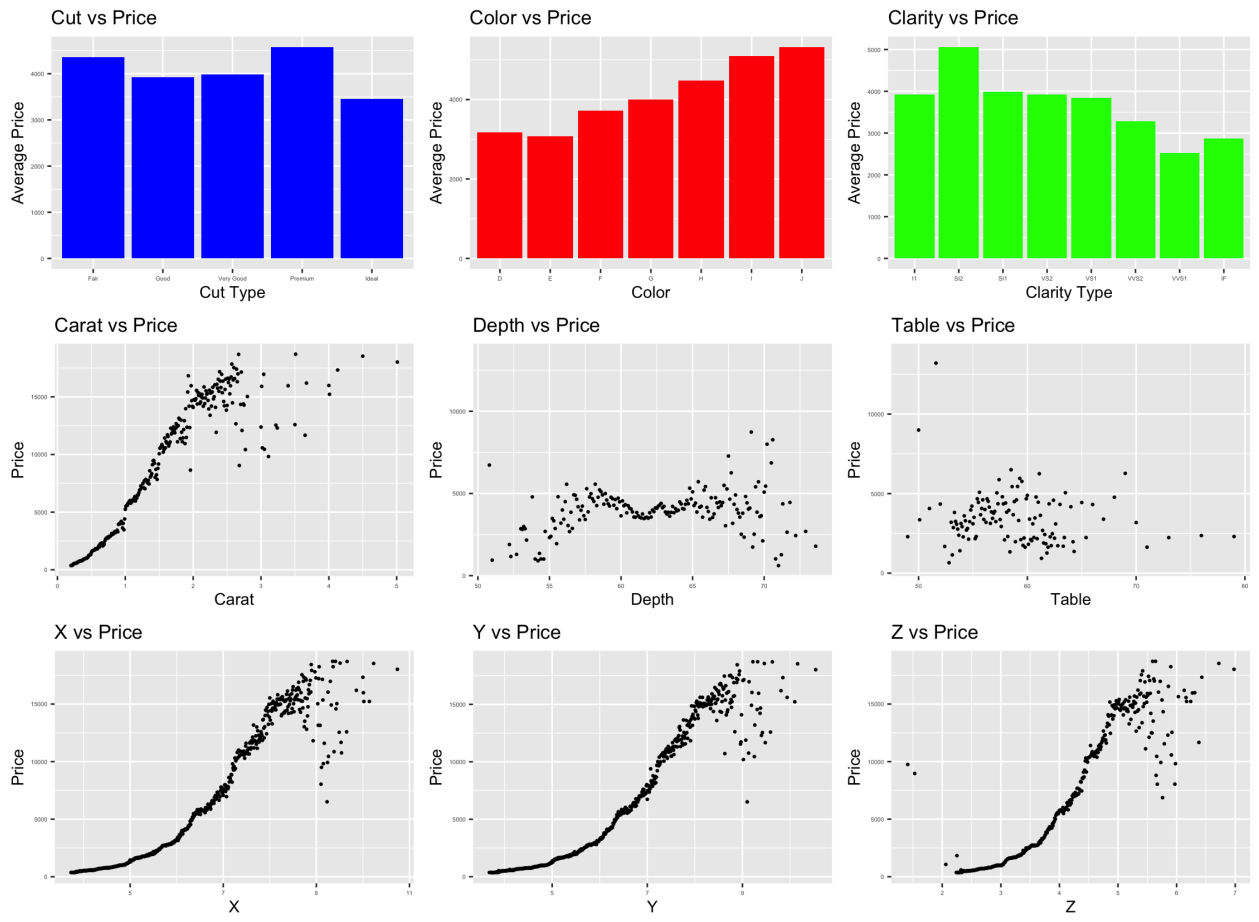

Figure 2: Relationships between variables

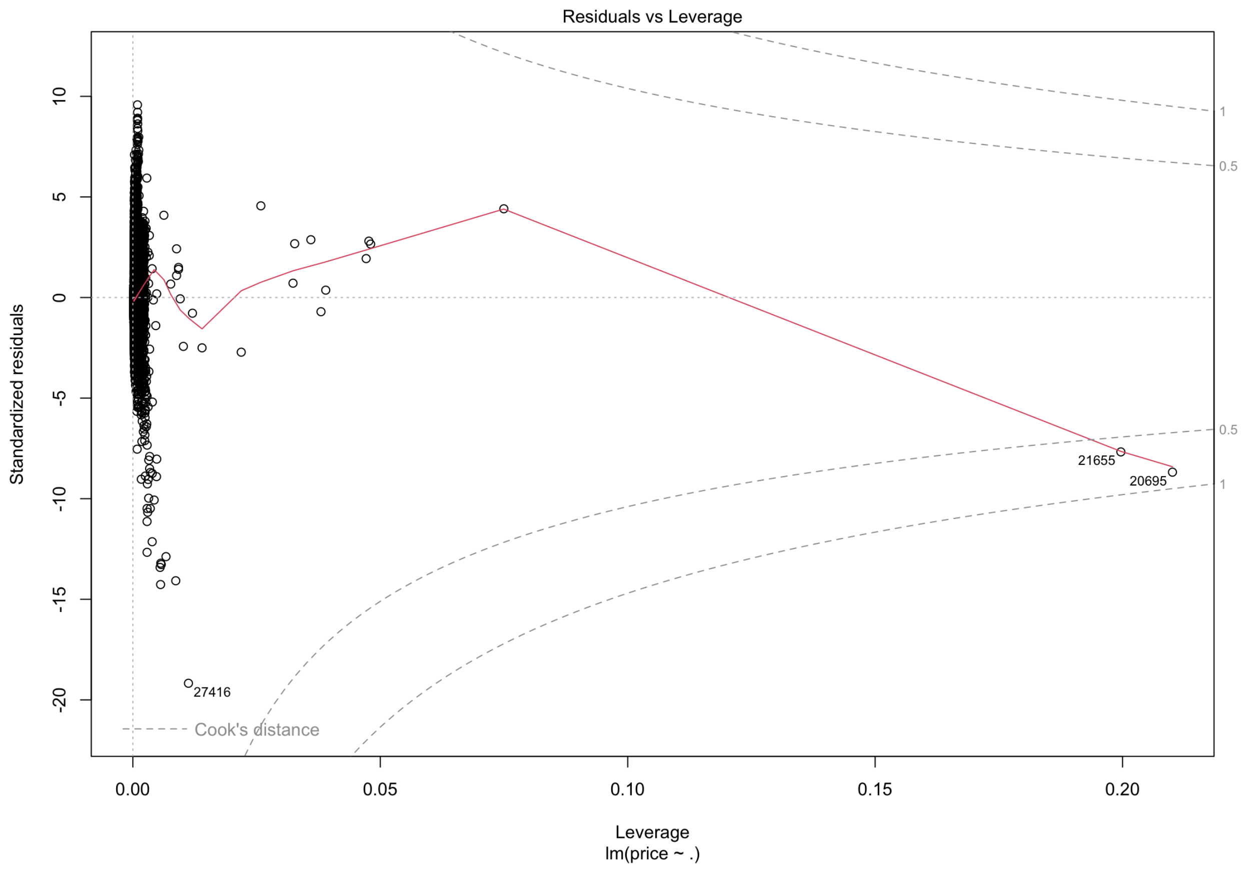

Figure 3: Residuals vs Leverage of Model 1

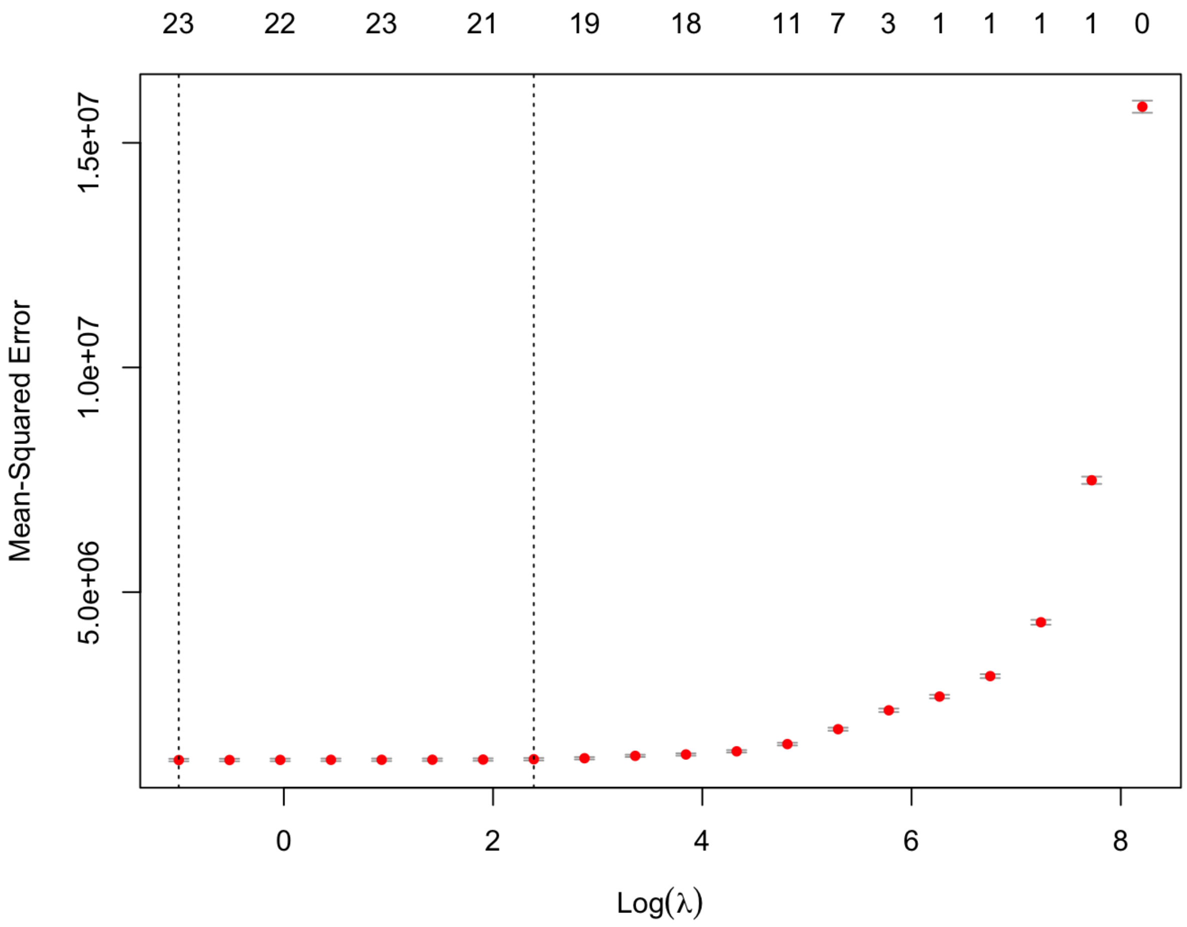

Figure 4: 10-fold CV MSE versus λ for Lasso regression

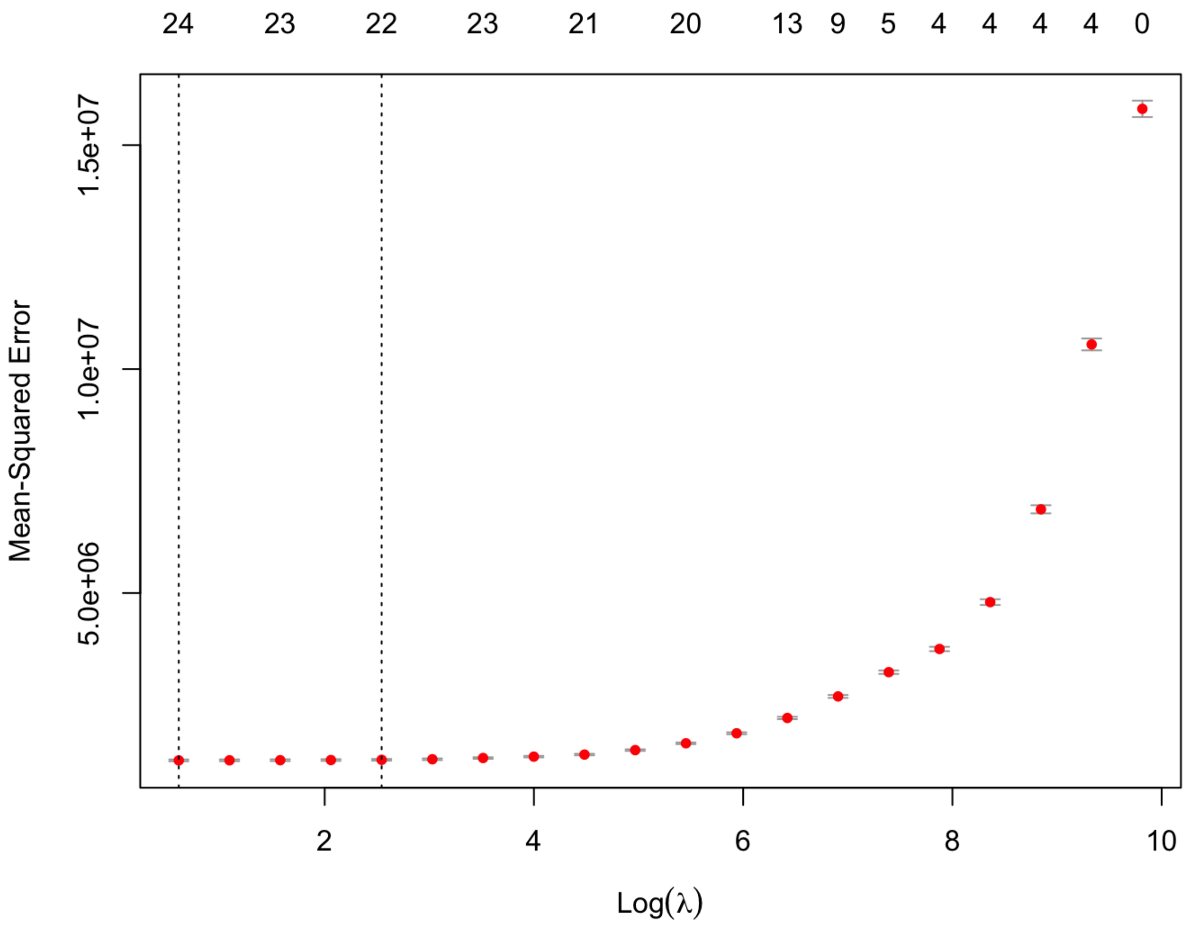

Figure 5: 10-fold CV MSE versus λ for Ridge regression