Final Project

Adib Osmany

2023-12-18

#Introduction:

Laptops are among the most versatile and popular personal computers in the world, with uses in education, business, entertainment, and more. You might be using one right now, as the internet has become an essential part of our lives in the era of artificial intelligence, social media, and online content. However, not everyone can afford a laptop, as their prices have been fluctuating over the years. Inflation is one of the reasons, but not the only one. Features and specifications also affect the cost of laptops, as technology advances and offers more options and capabilities. With this in mind, one question I aim to answer is if we can build a prediction model based on different types of hardware and features from a laptop to predict its price. With this information we could be able to predict future prices of laptops. One real world application of this information would be to use this regression model in government programs designed to issue access to the internet to low-income households by handing out affordable laptops with decent specs. Another real world application would simply just be using it for our own personal use to check if the contents of a specific laptop is worth the price compared to laptops with slightly different specs.

#Data Description

The source of all of my data comes from a website called kaggle which is run by a data science company. The data set I picked had over 1000 inputs, 1273 to be exact. The original data set contained 13 variables. These variables include company from which the laptop was made, Type of laptop, amount of Ram, the weight of the laptop, if it is touch screen or not, if it has an ips screen, what the the PPS, the CPU brand, the size of its HDD, the size of its SSD, the GPU brand, the operating system, and of course the price, which is the variable that we will try to predict. Here is what the data set looks like before any modifications.

Company <chr>	**TypeName** <chr>	**Ram** <int>	**Weight** <dbl>	**Price** <dbl>	**TouchScreen** <int>	**Ips** <int>	**Ppi** <dbl>	**Cpu_brand** <chr>	**HDD** <int>	**SSD** <int>
Apple	Ultrabook	8	1.370	11.175755	0	1	226.9830	Intel Core i5	0	128

Apple	Ultrabook	8	1.340	10.776777	0	0	127.6779	Intel Core i5	0	0
HP	Notebook	8	1.860	10.329931	0	0	141.2120	Intel Core i5	0	256
Apple	Ultrabook	16	1.830	11.814476	0	1	220.5346	Intel Core i7	0	512
Apple	Ultrabook	8	1.370	11.473101	0	1	226.9830	Intel Core i5	0	256
Acer	Notebook	4	2.100	9.967026	0	0	100.4547	AMD Processor	500	0
Apple	Ultrabook	16	2.040	11.644108	0	1	220.5346	Intel Core i7	0	0
Apple	Ultrabook	8	1.340	11.030615	0	0	127.6779	Intel Core i5	0	0
Asus	Ultrabook	16	1.300	11.285443	0	0	157.3505	Intel Core i7	0	512
Acer	Ultrabook	8	1.600	10.621952	0	1	157.3505	Intel Core i5	0	256
HP	Notebook	4	1.860	9.951658	0	0	100.4547	Intel Core i5	500	0

While the data set is vast with many data inputs, some of the categories would be difficult to use when creating a linear regression model. These categories include the company, the type of laptop, the CPU brand, the GPU brand and the OS. These are all categories to non-numerical values. One solution I researched about are dummy variables. This is when we take the unique inputs in each category and make a whole new variable out of them. This variable could only be represented with the number 1 or 0. Since these variables would only be represented in binary, it is up to R to come up with a good enough Bi constant to represent these inputs. I went about cleaning the data using python. With the help of youtube, I was able to learn how to access and manipulate data from a data set. My first step was to download and import a library called pandas into my python file. This pandas extension allows people to access csv files, which is the file type of google sheets and microsoft spreadsheets. After gaining access to a copy of the data set, I coded a for-loop that put all of the non-numerical categories into a list. Using this list, I made another list which contained lists of all the unique objects in each of the non-numerical categories. I called this list objList. Using another for loop, I took each element in the objList list and created new columns for each object. While doing this, I had my program fill in the cells with either True or False, represented in the form of '1' or '0', respectfully. True meant that this certain laptop had this feature. False meant that it did not. After all of the new columns were created, I went and deleted the columns I no longer needed, this being all the non-numerical columns. Here is what the data set looks like after modifications.

Ram <int>	Weight <dbl>	Price <dbl>	TouchScreen <int>	Ips <int>	Ppi <dbl>	HDD <int>	SSD <int>	is_Apple <int>	is_HP <int>	is_Acer <int>	is_Asus <int>	is_Dell <int>	is_Lenovo <int>
8	1.370	11.175755	0	1	226.9830	0	128	1	0	0	0	0	0
8	1.340	10.776777	0	0	127.6779	0	0	1	0	0	0	0	0
8	1.860	10.329931	0	0	141.2120	0	256	0	1	0	0	0	0
16	1.830	11.814476	0	1	220.5346	0	512	1	0	0	0	0	0
8	1.370	11.473101	0	1	226.9830	0	256	1	0	0	0	0	0
4	2.100	9.967026	0	0	100.4547	500	0	0	0	1	0	0	0
16	2.040	11.644108	0	1	220.5346	0	0	1	0	0	0	0	0
8	1.340	11.030615	0	0	127.6779	0	0	1	0	0	0	0	0
16	1.300	11.285443	0	0	157.3505	0	512	0	0	0	1	0	0
8	1.600	10.621952	0	1	157.3505	0	256	0	0	1	0	0	0
4	1.860	9.951658	0	0	100.4547	500	0	0	1	0	0	0	0

Once I was able to see this new dataset myself, I realized that there may have been too many variables to work with. However, I still continued with this data set since it had almost all of the variables I was looking for; but a note to my future self would probably be to take some extra time to pick a data set that doesn't have non-numerical categories as to avoid the pain of turning them into dummy variables.

#Methods

The first step I took was uploading my cleaned csv file into R studio. After that, I took my 5 non-numerical categories and made lists for all of them for any variables that belonged to said category. Ex:

```
## [1] "Company"

##  [1] "data$is_Apple"     "data$is_HP"        "data$is_Acer"
##  [4] "data$is_Asus"      "data$is_Dell"      "data$is_Lenovo"
##  [7] "data$is_Chuwi"     "data$is_MSI"       "data$is_Microsoft"
## [10] "data$is_Toshiba"   "data$is_Huawei"    "data$is_Xiaomi"
## [13] "data$is_Vero"      "data$is_Razer"     "data$is_Mediacom"
## [16] "data$is_Samsung"   "data$is_Google"    "data$is_Fujitsu"
## [19] "data$is_LG"

## [1] "CPU"

## [1] "data$is_Intel.Core.i5"         "data$is_Intel.Core.i7"
## [3] "data$is_AMD.Processor"         "data$is_Intel.Core.i3"
## [5] "data$is_Other.Intel.Processor"

## [1] "GPU"

## [1] "data$is_Intel"  "data$is_AMD"    "data$is_Nvidia"

## [1] "OS"

## [1] "data$is_Mac"     "data$is_Others"  "data$is_Windows"
```

After I sorted my variables into their proper categories, I went ahead and made a linear regression function that would help me create models using these non-numerical categories. With this function I went ahead and created my first model. This model consisted of all the variables as predictors. The reason I decided to use all of the predictors is so that I could get a general idea of how my other models should look like. I also just wanted to make sure that these predictors are in the slightest bit helpful. The answer to that is yes, as the model had a relatively high F-test, had a p-value less than ,05, and had a an R^2adj value relatively close to 1, and was only slightly different from the regular R^2 value, which tells me that the predictors are good independent variables for this model. After analyzing this model, which I called "full_model," I went ahead and decided to create three other models using the model selection algorithms. These include backward elimination, forward selection, and bit wise regression. I analyzed each of these models and made a guess as to which of these models would be the best. However, before I went on to the anova table. I created one last model. This model was merely a model I wanted to test out myself using the data I analyzed in the previous to see if I could make a better model. All these models and results will be available in the "Results" section. However, before we jump into the results, there were some issues that arose while doing this project that would be a good time to clear up. One issue being that some of the predictors have 'NA' written on them. This issue was never properly resolved but it poses no harm as the model would just ignore it as a predictor all together. Another issue that arose is trying to figure out how to organize the variables into the categories. Luckily, with the power of the internet, I was able to figure out a solution, which was to use lists. I was also able to figure out how to make the linear regression function work with certain inputs. Another issue I was able to find a solution for was an easy way to implement the model selection algorithms. Luckily I was able to find a function to this and make new models to be used.

#Results

Lets begin with my model of all the predictors:

```
##
## Call:
## lm(formula = formula)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85369 -0.17048 -0.01246  0.15570  1.04641
##
## Coefficients: (6 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.044e+01  1.700e-01  61.403  < 2e-16 ***
## data$is_Apple                 -1.566e-01  1.662e-01  -0.942 0.346198
## data$is_HP                    -2.636e-01  1.552e-01  -1.698 0.089788 .
## data$is_Acer                  -4.630e-01  1.568e-01  -2.953 0.003210 **
## data$is_Asus                  -3.668e-01  1.562e-01  -2.348 0.019018 *
## data$is_Dell                  -3.068e-01  1.553e-01  -1.976 0.048394 *
## data$is_Lenovo                -3.447e-01  1.549e-01  -2.224 0.026303 *
## data$is_Chuwi                 -8.261e-01  2.190e-01  -3.773 0.000169 ***
## data$is_MSI                   -2.319e-01  1.607e-01  -1.443 0.149235
## data$is_Microsoft             -5.773e-02  1.865e-01  -0.310 0.756930
## data$is_Toshiba               -1.537e-01  1.579e-01  -0.973 0.330587
## data$is_Huawei                -4.133e-01  2.394e-01  -1.727 0.084479 .
## data$is_Xiaomi                -2.376e-01  2.030e-01  -1.170 0.242123
## data$is_Vero                  -1.009e+00  2.042e-01  -4.940 8.89e-07 ***
## data$is_Razer                 -2.198e-01  1.846e-01  -1.191 0.233915
## data$is_Mediacom              -8.112e-01  1.856e-01  -4.370 1.35e-05 ***
## data$is_Samsung               -1.505e-01  1.796e-01  -0.838 0.402135
## data$is_Google                 4.910e-03  2.165e-01   0.023 0.981910
## data$is_Fujitsu               -4.402e-01  2.157e-01  -2.041 0.041438 *
## data$is_LG                            NA         NA      NA       NA
## data$is_Ultrabook             -3.881e-01  5.842e-02  -6.643 4.59e-11 ***
## data$is_Notebook              -6.448e-01  5.309e-02 -12.146  < 2e-16 ***
## data$is_Netbook               -6.217e-01  7.846e-02  -7.924 5.11e-15 ***
## data$is_Gaming                -4.255e-01  5.613e-02  -7.581 6.73e-14 ***
## data$is_2.in.1.Convertible    -3.955e-01  6.803e-02  -5.814 7.75e-09 ***
## data$is_Workstation                   NA         NA      NA       NA
## data$is_Intel.Core.i5          5.246e-01  2.964e-02  17.701  < 2e-16 ***
## data$is_Intel.Core.i7          5.872e-01  3.327e-02  17.649  < 2e-16 ***
## data$is_AMD.Processor          1.177e-01  5.219e-02   2.256 0.024267 *
## data$is_Intel.Core.i3          2.212e-01  3.459e-02   6.395 2.27e-10 ***
## data$is_Other.Intel.Processor         NA         NA      NA       NA
## data$is_Intel                 -2.205e-03  2.525e-02  -0.087 0.930442
## data$is_AMD                   -8.820e-02  3.293e-02  -2.678 0.007496 **
## data$is_Nvidia                        NA         NA      NA       NA
## data$is_Mac                           NA         NA      NA       NA
## data$is_Others                -2.369e-01  2.477e-02  -9.562  < 2e-16 ***
## data$is_Windows                          NA         NA      NA       NA    
## data$Ram                          2.613e-02  2.439e-03  10.716  < 2e-16 ***
## data$Weight                       7.267e-02  1.991e-02   3.651 0.000272 ***
## data$Ips                          3.695e-02  1.960e-02   1.885 0.059661 .  
## data$HDD                         -1.297e-06  2.021e-05  -0.064 0.948818    
## data$SSD                          5.795e-04  6.726e-05   8.617  < 2e-16 ***
## data$Ppi                          2.489e-03  2.480e-04  10.038  < 2e-16 ***
## data$TouchScreen                 -1.049e-01  3.851e-02  -2.724 0.006545 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2594 on 1235 degrees of freedom
## Multiple R-squared:  0.8298, Adjusted R-squared:  0.8247 
## F-statistic: 162.8 on 37 and 1235 DF,  p-value: < 2.2e-16
```

As you can see from this model, the F-test is greater than .05 and the p-value is less than .05, making this model a good predictor. The R^2 adjusted is also close to 1, which means that most of the predictors are well used as predictors. This model serves as a base model to be compared to the other models, however that does not mean that this model will end up being the worst out of the other models.

Our next model is the one we use backward selection on:

```
## 
## Call:
## lm(formula = data$Price ~ data$is_HP + data$is_Acer + data$is_Asus + 
##     data$is_Dell + data$is_Lenovo + data$is_Chuwi + data$is_MSI + 
##     data$is_Huawei + data$is_Vero + data$is_Mediacom + data$is_Fujitsu + 
##     data$is_Ultrabook + data$is_Notebook + data$is_Netbook + 
##     data$is_Gaming + data$is_2.in.1.Convertible + data$is_Intel.Core.i5 + 
##     data$is_Intel.Core.i7 + data$is_AMD.Processor + data$is_Intel.Core.i3 
+ 
##     data$is_AMD + data$is_Others + data$Ram + data$Weight + data$Ips + 
##     data$SSD + data$Ppi + data$TouchScreen)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.84716 -0.17010 -0.01078  0.15554  1.04986 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  1.030e+01  8.163e-02 126.135  < 2e-16 ***
## data$is_HP                  -1.140e-01  3.261e-02  -3.497 0.000486 ***
## data$is_Acer                -3.133e-01  3.976e-02  -7.879 7.18e-15 ***
## data$is_Asus                -2.163e-01  3.641e-02  -5.941 3.67e-09 ***
## data$is_Dell                -1.585e-01  3.264e-02  -4.856 1.35e-06 ***
## data$is_Lenovo              -1.946e-01  3.247e-02  -5.994 2.68e-09 ***
## data$is_Chuwi               -6.722e-01  1.549e-01  -4.341 1.54e-05 ***
## data$is_MSI                 -8.001e-02  5.158e-02  -1.551 0.121104    
## data$is_Huawei              -2.649e-01  1.857e-01  -1.426 0.154090    
## data$is_Vero              -8.567e-01  1.352e-01  -6.336 3.29e-10 ***
## data$is_Mediacom          -6.586e-01  1.056e-01  -6.235 6.18e-10 ***
## data$is_Fujitsu           -2.914e-01  1.529e-01  -1.906 0.056920 .
## data$is_Ultrabook         -3.863e-01  5.584e-02  -6.919 7.28e-12 ***
## data$is_Notebook          -6.478e-01  5.137e-02 -12.611  < 2e-16 ***
## data$is_Netbook           -6.249e-01  7.711e-02  -8.104 1.26e-15 ***
## data$is_Gaming            -4.280e-01  5.515e-02  -7.760 1.76e-14 ***
## data$is_2.in.1.Convertible -4.113e-01  6.498e-02  -6.330 3.42e-10 ***
## data$is_Intel.Core.i5      5.256e-01  2.906e-02  18.088  < 2e-16 ***
## data$is_Intel.Core.i7      5.892e-01  3.169e-02  18.592  < 2e-16 ***
## data$is_AMD.Processor      1.180e-01  5.167e-02   2.284 0.022536 *
## data$is_Intel.Core.i3      2.212e-01  3.426e-02   6.457 1.52e-10 ***
## data$is_AMD               -8.600e-02  2.831e-02  -3.038 0.002431 **
## data$is_Others            -2.350e-01  2.409e-02  -9.755  < 2e-16 ***
## data$Ram                   2.584e-02  2.376e-03  10.876  < 2e-16 ***
## data$Weight                7.307e-02  1.884e-02   3.878 0.000111 ***
## data$Ips                   3.663e-02  1.900e-02   1.928 0.054036 .
## data$SSD                   5.827e-04  5.812e-05  10.026  < 2e-16 ***
## data$Ppi                   2.449e-03  2.386e-04  10.263  < 2e-16 ***
## data$TouchScreen          -8.877e-02  3.566e-02  -2.489 0.012937 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2588 on 1244 degrees of freedom
## Multiple R-squared:  0.8294, Adjusted R-squared:  0.8255
## F-statistic: 215.9 on 28 and 1244 DF,  p-value: < 2.2e-16
```

From this model we see that the predictors used are Company, touch screen, ppi, ssd, ips, ram, weight, laptop type, cpu, and gpu. Compared to our first model, we see that this model has a higher F-value, the same p-value, and a higher R^2 adj value. Based on this information we could see that this model is better than our first model.

Forward selection model:

```
##
## Call:
## lm(formula = data$Price ~ data$is_Apple + data$is_HP + data$is_Acer +
##     data$is_Asus + data$is_Dell + data$is_Lenovo + data$is_Chuwi +
##     data$is_MSI + data$is_Microsoft + data$is_Toshiba + data$is_Huawei +
##     data$is_Xiaomi + data$is_Vero + data$is_Razer + data$is_Mediacom +
##     data$is_Samsung + data$is_Google + data$is_Fujitsu + data$is_LG +
##     data$is_Ultrabook + data$is_Notebook + data$is_Netbook +
##     data$is_Gaming + data$is_2.in.1.Convertible + data$is_Workstation +
##     data$is_Intel.Core.i5 + data$is_Intel.Core.i7 + data$is_AMD.Processor
+
##     data$is_Intel.Core.i3 + data$is_Other.Intel.Processor + data$is_Intel
+
##     data$is_AMD + data$is_Nvidia + data$is_Mac + data$is_Others +
##     data$is_Windows + data$Ram + data$Weight + data$Ips + data$HDD +
##     data$SSD + data$Ppi + data$TouchScreen)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85369 -0.17048 -0.01246  0.15570  1.04641
##
## Coefficients: (6 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.044e+01  1.700e-01  61.403  < 2e-16 ***
## data$is_Apple                -1.566e-01  1.662e-01  -0.942 0.346198
## data$is_HP                   -2.636e-01  1.552e-01  -1.698 0.089788 .
## data$is_Acer                 -4.630e-01  1.568e-01  -2.953 0.003210 **
## data$is_Asus                 -3.668e-01  1.562e-01  -2.348 0.019018 *
## data$is_Dell                 -3.068e-01  1.553e-01  -1.976 0.048394 *
## data$is_Lenovo               -3.447e-01  1.549e-01  -2.224 0.026303 *
## data$is_Chuwi                -8.261e-01  2.190e-01  -3.773 0.000169 ***
## data$is_MSI                  -2.319e-01  1.607e-01  -1.443 0.149235
## data$is_Microsoft            -5.773e-02  1.865e-01  -0.310 0.756930
## data$is_Toshiba              -1.537e-01  1.579e-01  -0.973 0.330587
## data$is_Huawei               -4.133e-01  2.394e-01  -1.727 0.084479 .
## data$is_Xiaomi               -2.376e-01  2.030e-01  -1.170 0.242123
## data$is_Vero                 -1.009e+00  2.042e-01  -4.940 8.89e-07 ***
## data$is_Razer                -2.198e-01  1.846e-01  -1.191 0.233915
## data$is_Mediacom             -8.112e-01  1.856e-01  -4.370 1.35e-05 ***
## data$is_Samsung              -1.505e-01  1.796e-01  -0.838 0.402135
## data$is_Google                4.910e-03  2.165e-01   0.023 0.981910
## data$is_Fujitsu              -4.402e-01  2.157e-01  -2.041 0.041438 *
## data$is_LG                           NA         NA      NA       NA
## data$is_Ultrabook            -3.881e-01  5.842e-02  -6.643 4.59e-11 ***
## data$is_Notebook             -6.448e-01  5.309e-02 -12.146  < 2e-16 ***
## data$is_Netbook              -6.217e-01  7.846e-02  -7.924 5.11e-15 ***
## data$is_Gaming               -4.255e-01  5.613e-02  -7.581 6.73e-14 ***
## data$is_2.in.1.Convertible   -3.955e-01  6.803e-02  -5.814 7.75e-09 ***
## data$is_Workstation                  NA         NA      NA       NA
## data$is_Intel.Core.i5         5.246e-01  2.964e-02  17.701  < 2e-16 ***
## data$is_Intel.Core.i7         5.872e-01  3.327e-02  17.649  < 2e-16 ***
## data$is_AMD.Processor         1.177e-01  5.219e-02   2.256 0.024267 *
## data$is_Intel.Core.i3         2.212e-01  3.459e-02   6.395 2.27e-10 ***
## data$is_Other.Intel.Processor        NA         NA      NA       NA
## data$is_Intel                -2.205e-03  2.525e-02  -0.087 0.930442
## data$is_AMD                  -8.820e-02  3.293e-02  -2.678 0.007496 **
## data$is_Nvidia                       NA         NA      NA       NA
## data$is_Mac                          NA         NA      NA       NA
## data$is_Others               -2.369e-01  2.477e-02  -9.562  < 2e-16 ***
## data$is_Windows                      NA         NA      NA       NA
## data$Ram                      2.613e-02  2.439e-03  10.716  < 2e-16 ***
## data$Weight                   7.267e-02  1.991e-02   3.651 0.000272 ***
## data$Ips                      3.695e-02  1.960e-02   1.885 0.059661 .
## data$HDD                     -1.297e-06  2.021e-05  -0.064 0.948818
## data$SSD                      5.795e-04  6.726e-05   8.617  < 2e-16 ***
## data$Ppi                      2.489e-03  2.480e-04  10.038  < 2e-16 ***
## data$TouchScreen              -1.049e-01  3.851e-02  -2.724 0.006545 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2594 on 1235 degrees of freedom
## Multiple R-squared:  0.8298, Adjusted R-squared:  0.8247
## F-statistic: 162.8 on 37 and 1235 DF,  p-value: < 2.2e-16
```

Using this forward selection algorithm, we see that we get the same results as the first model. What this can tell us is that almost all of the predictors are significant. Meaning there is a predictor that isn’t independent and helps to predict the price. While this can sometimes be a good thing, there can also be cases where having too many predictors actually skews your data too much, this is why it is best to find the best fit model using the best predictors.

Our final model selection algorithm model is the stepwise model:

```
##
## Call:
## lm(formula = data$Price ~ data$is_HP + data$is_Acer + data$is_Asus +
##     data$is_Dell + data$is_Lenovo + data$is_Chuwi + data$is_MSI +
##     data$is_Huawei + data$is_Vero + data$is_Mediacom + data$is_Fujitsu +
##     data$is_Ultrabook + data$is_Notebook + data$is_Netbook +
##     data$is_Gaming + data$is_2.in.1.Convertible + data$is_Intel.Core.i5 +
##     data$is_Intel.Core.i7 + data$is_AMD.Processor + data$is_Intel.Core.i3
+
##     data$is_AMD + data$is_Others + data$Ram + data$Weight + data$Ips +
##     data$SSD + data$Ppi + data$TouchScreen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84716 -0.17010 -0.01078  0.15554  1.04986
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.030e+01  8.163e-02 126.135  < 2e-16 ***
## data$is_HP                 -1.140e-01  3.261e-02  -3.497 0.000486 ***
## data$is_Acer               -3.133e-01  3.976e-02  -7.879 7.18e-15 ***
## data$is_Asus               -2.163e-01  3.641e-02  -5.941 3.67e-09 ***
## data$is_Dell               -1.585e-01  3.264e-02  -4.856 1.35e-06 ***
## data$is_Lenovo             -1.946e-01  3.247e-02  -5.994 2.68e-09 ***
## data$is_Chuwi              -6.722e-01  1.549e-01  -4.341 1.54e-05 ***
## data$is_MSI                -8.001e-02  5.158e-02  -1.551 0.121104
## data$is_Huawei             -2.649e-01  1.857e-01  -1.426 0.154090
## data$is_Vero               -8.567e-01  1.352e-01  -6.336 3.29e-10 ***
## data$is_Mediacom           -6.586e-01  1.056e-01  -6.235 6.18e-10 ***
## data$is_Fujitsu            -2.914e-01  1.529e-01  -1.906 0.056920 .
## data$is_Ultrabook          -3.863e-01  5.584e-02  -6.919 7.28e-12 ***
## data$is_Notebook           -6.478e-01  5.137e-02 -12.611  < 2e-16 ***
## data$is_Netbook            -6.249e-01  7.711e-02  -8.104 1.26e-15 ***
## data$is_Gaming            -4.280e-01  5.515e-02  -7.760 1.76e-14 ***
## data$is_2.in.1.Convertible -4.113e-01  6.498e-02  -6.330 3.42e-10 ***
## data$is_Intel.Core.i5       5.256e-01  2.906e-02  18.088  < 2e-16 ***
## data$is_Intel.Core.i7       5.892e-01  3.169e-02  18.592  < 2e-16 ***
## data$is_AMD.Processor       1.180e-01  5.167e-02   2.284 0.022536 *
## data$is_Intel.Core.i3       2.212e-01  3.426e-02   6.457 1.52e-10 ***
## data$is_AMD                -8.600e-02  2.831e-02  -3.038 0.002431 **
## data$is_Others             -2.350e-01  2.409e-02  -9.755  < 2e-16 ***
## data$Ram                    2.584e-02  2.376e-03  10.876  < 2e-16 ***
## data$Weight                 7.307e-02  1.884e-02   3.878 0.000111 ***
## data$Ips                    3.663e-02  1.900e-02   1.928 0.054036 .
## data$SSD                    5.827e-04  5.812e-05  10.026  < 2e-16 ***
## data$Ppi                    2.449e-03  2.386e-04  10.263  < 2e-16 ***
## data$TouchScreen           -8.877e-02  3.566e-02  -2.489 0.012937 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2588 on 1244 degrees of freedom
## Multiple R-squared:  0.8294, Adjusted R-squared:  0.8255
## F-statistic: 215.9 on 28 and 1244 DF,  p-value: < 2.2e-16
```

From this model we know that it is exactly the same as our backward selection model. What this might tell us is that the stepwise algorithm most likely implemented the forward selection algorithm first to get all the predictors, then it used the backward selection algorithm to root out any unnecessary predictors.

So far we have 2 sets of identical models. The last model I wish to compare is my own personal model. I made this model just as a way to see if I could create a better model than the algorithms we used.

```
##
## Call:
## lm(formula = formula)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89130 -0.17620 -0.01184  0.16534  1.05935
##
## Coefficients: (3 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.036e+01  1.745e-01  59.398  < 2e-16 ***
## data$Ram                      2.645e-02  2.492e-03  10.613  < 2e-16 ***
## data$is_Intel.Core.i5         5.608e-01  3.002e-02  18.678  < 2e-16 ***
## data$is_Intel.Core.i7         6.143e-01  3.284e-02  18.707  < 2e-16 ***
## data$is_AMD.Processor         1.010e-01  4.434e-02   2.278 0.022913 *
## data$is_Intel.Core.i3         2.414e-01  3.561e-02   6.778 1.88e-11 ***
## data$is_Other.Intel.Processor        NA         NA      NA       NA
## data$TouchScreen             -8.133e-02  3.982e-02  -2.042 0.041354 *
## data$Ips                      4.929e-02  2.029e-02   2.429 0.015272 *
## data$Ppi                      2.513e-03  2.565e-04   9.799  < 2e-16 ***
## data$SSD                        6.393e-04  6.126e-05  10.436  < 2e-16 ***
## data$is_Ultrabook              -4.017e-01  5.864e-02  -6.850 1.16e-11 ***
## data$is_Notebook               -6.812e-01  5.342e-02 -12.752  < 2e-16 ***
## data$is_Netbook                -6.726e-01  8.036e-02  -8.370  < 2e-16 ***
## data$is_Gaming                 -4.385e-01  5.759e-02  -7.614 5.23e-14 ***
## data$is_2.in.1.Convertible    -4.294e-01  6.891e-02  -6.232 6.32e-10 ***
## data$is_Workstation                    NA         NA      NA       NA    
## data$Weight                     6.827e-02  1.970e-02   3.466 0.000547 ***
## data$is_Apple                  -1.294e-01  1.726e-01  -0.750 0.453526    
## data$is_HP                     -2.224e-01  1.614e-01  -1.378 0.168366    
## data$is_Acer                   -4.459e-01  1.629e-01  -2.737 0.006287 ** 
## data$is_Asus                   -3.314e-01  1.624e-01  -2.041 0.041472 *  
## data$is_Dell                   -2.997e-01  1.614e-01  -1.857 0.063558 .  
## data$is_Lenovo                 -3.345e-01  1.611e-01  -2.077 0.038047 *  
## data$is_Chuwi                  -7.176e-01  2.271e-01  -3.160 0.001614 ** 
## data$is_MSI                    -1.828e-01  1.670e-01  -1.095 0.273901    
## data$is_Microsoft              -3.602e-02  1.938e-01  -0.186 0.852595    
## data$is_Toshiba                -9.898e-02  1.640e-01  -0.604 0.546223    
## data$is_Huawei                 -3.949e-01  2.490e-01  -1.586 0.113036    
## data$is_Xiaomi                 -3.749e-01  2.091e-01  -1.793 0.073231 .  
## data$is_Vero                   -9.000e-01  2.118e-01  -4.248 2.31e-05 ***
## data$is_Razer                  -2.098e-01  1.917e-01  -1.094 0.274057    
## data$is_Mediacom               -7.132e-01  1.925e-01  -3.705 0.000221 ***
## data$is_Samsung                -1.370e-01  1.865e-01  -0.734 0.462847    
## data$is_Google                 -2.227e-01  2.238e-01  -0.995 0.320032    
## data$is_Fujitsu                -3.708e-01  2.241e-01  -1.655 0.098260 .  
## data$is_LG                             NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2699 on 1239 degrees of freedom
## Multiple R-squared:  0.8152, Adjusted R-squared:  0.8103 
## F-statistic: 165.6 on 33 and 1239 DF,  p-value: < 2.2e-16
```

This model uses ram,cpu,touch,ips,ppi,ssd,type,weight and company as the predictors. As can be seen, this model has a better F-value than the original model and forward selection model. The p-value seems to have remained the same all throughout. The R^2 squared value on the other hand is the worst out of all 5 models.

Now we use an anova table to compare:

```
## Analysis of Variance Table
## 
## Model 1: data$Price ~ data$is_Apple + data$is_HP + data$is_Acer + 
data$is_Asus + 
##     data$is_Dell + data$is_Lenovo + data$is_Chuwi + data$is_MSI + 
##     data$is_Microsoft + data$is_Toshiba + data$is_Huawei + data$is_Xiaomi 
+ 
##     data$is_Vero + data$is_Razer + data$is_Mediacom + data$is_Samsung + 
##     data$is_Google + data$is_Fujitsu + data$is_LG + data$is_Ultrabook + 
##     data$is_Notebook + data$is_Netbook + data$is_Gaming +
data$is_2.in.1.Convertible +
##     data$is_Workstation + data$is_Intel.Core.i5 + data$is_Intel.Core.i7 +
##     data$is_AMD.Processor + data$is_Intel.Core.i3 +
data$is_Other.Intel.Processor +
##     data$is_Intel + data$is_AMD + data$is_Nvidia + data$is_Mac +
##     data$is_Others + data$is_Windows + data$Ram + data$Weight +
##     data$Ips + data$HDD + data$SSD + data$Ppi + data$TouchScreen
## Model 2: data$Price ~ data$is_HP + data$is_Acer + data$is_Asus +
data$is_Dell +
##     data$is_Lenovo + data$is_Chuwi + data$is_MSI + data$is_Huawei +
##     data$is_Vero + data$is_Mediacom + data$is_Fujitsu + data$is_Ultrabook
+
##     data$is_Notebook + data$is_Netbook + data$is_Gaming +
data$is_2.in.1.Convertible +
##     data$is_Intel.Core.i5 + data$is_Intel.Core.i7 + data$is_AMD.Processor
+
##     data$is_Intel.Core.i3 + data$is_AMD + data$is_Others + data$Ram +
##     data$Weight + data$Ips + data$SSD + data$Ppi + data$TouchScreen
## Model 3: data$Price ~ data$is_Apple + data$is_HP + data$is_Acer +
data$is_Asus +
##     data$is_Dell + data$is_Lenovo + data$is_Chuwi + data$is_MSI +
##     data$is_Microsoft + data$is_Toshiba + data$is_Huawei + data$is_Xiaomi
+
##     data$is_Vero + data$is_Razer + data$is_Mediacom + data$is_Samsung +
##     data$is_Google + data$is_Fujitsu + data$is_LG + data$is_Ultrabook +
##     data$is_Notebook + data$is_Netbook + data$is_Gaming +
data$is_2.in.1.Convertible +
##     data$is_Workstation + data$is_Intel.Core.i5 + data$is_Intel.Core.i7 +
##     data$is_AMD.Processor + data$is_Intel.Core.i3 +
data$is_Other.Intel.Processor +
##     data$is_Intel + data$is_AMD + data$is_Nvidia + data$is_Mac +
##     data$is_Others + data$is_Windows + data$Ram + data$Weight +
##     data$Ips + data$HDD + data$SSD + data$Ppi + data$TouchScreen
## Model 4: data$Price ~ data$is_HP + data$is_Acer + data$is_Asus +
data$is_Dell +
##     data$is_Lenovo + data$is_Chuwi + data$is_MSI + data$is_Huawei +
##     data$is_Vero + data$is_Mediacom + data$is_Fujitsu + data$is_Ultrabook
+
##     data$is_Notebook + data$is_Netbook + data$is_Gaming +
data$is_2.in.1.Convertible +
##     data$is_Intel.Core.i5 + data$is_Intel.Core.i7 + data$is_AMD.Processor
+
##     data$is_Intel.Core.i3 + data$is_AMD + data$is_Others + data$Ram +
##     data$Weight + data$Ips + data$SSD + data$Ppi + data$TouchScreen
## Model 5: data$Price ~ data$Ram + data$is_Intel.Core.i5 +
data$is_Intel.Core.i7 +
##     data$is_AMD.Processor + data$is_Intel.Core.i3 +
data$is_Other.Intel.Processor +
##     data$TouchScreen + data$Ips + data$Ppi + data$SSD + data$is_Ultrabook
```

```
+
##     data$is_Notebook + data$is_Netbook + data$is_Gaming +
data$is_2.in.1.Convertible +
##     data$is_Workstation + data$Weight + data$is_Apple + data$is_HP +
##     data$is_Acer + data$is_Asus + data$is_Dell + data$is_Lenovo +
##     data$is_Chuwi + data$is_MSI + data$is_Microsoft + data$is_Toshiba +
##     data$is_Huawei + data$is_Xiaomi + data$is_Vero + data$is_Razer +
##     data$is_Mediacom + data$is_Samsung + data$is_Google + data$is_Fujitsu
+
##     data$is_LG
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1235 83.083
## 2   1244 83.317 -9   -0.2339 0.3863 0.9421
## 3   1235 83.083  9    0.2339 0.3863 0.9421
## 4   1244 83.317 -9   -0.2339 0.3863 0.9421
## 5   1239 90.223  5   -6.9059
```

From the anova table we see that the best model we could possibly use is the original or forward selection model, which we already established are the same model. This is because they have the lowest RSS value. The backward model and the step wise, which we originally thought would be better, is only slightly higher. The model that I made is actually much higher than the rest. One thing I can take away from this is that not to let f-tests and p-values fool you. Another thing I could take away is to trust the algorithms and predictors given to us instead of trying to figure out my own model.

#Results

After participating in this project I learned many things that I probably would not have learned through a textbook. One of these things being judging when a model is good or at least better than another model. I believe I made the misjudgment about which model is earlier because of how close the models were to each other. In fact I believe the models have enough close resemblance that anyone could pick any of the models, excluding my personal model, and still achieve a good enough price prediction. I also learned that Kaggle is an amazing place to find data sets, so if I ever work on a project like this again, I will be sure to use it again next time. I also believe that we have successfully answered our question from the introduction, which is if we could build a good prediction model with data about hardware and software. The answer is yes, we saw that our models performed pretty well. However, I believe with a bigger and more diverse data set, we can build an even better model. In conclusion, this project taught me to trust the linear regression process, and to keep digging for my data, because it will help in the long run.