**DATS 6313 – Time Series Analysis & Modeling**

Instructor: Reza Jafari

**Lab #5**

Bradley Reardon

2/23/2022

**Abstract:**

This lab pertains to implementing feature reduction and various regression modeling techniques:

* Feature Reduction:
  + Singular Value Decomposition (SVD)
  + Backward Stepwise regression
* Multiple Linear Regression Models:
  + Least Squares Error (LSE)
  + Ordinary Least Squares (OLS)

The dataset used in this lab can be found [here](https://github.com/rjafari979/Time-Series-Analysis-and-Moldeing).

**Introduction:**

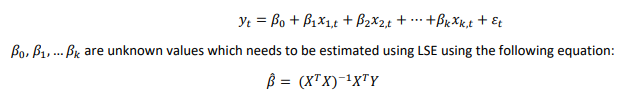
This experiment was performed to increase understanding of the application of two types of multiple linear regression models: LSE and OLS. The results of the two methods were compared and are displayed below. Additionally, this experiment required using both SVD and backward stepwise regression as methods for feature reduction.

**Method, Theory, and Procedures:**

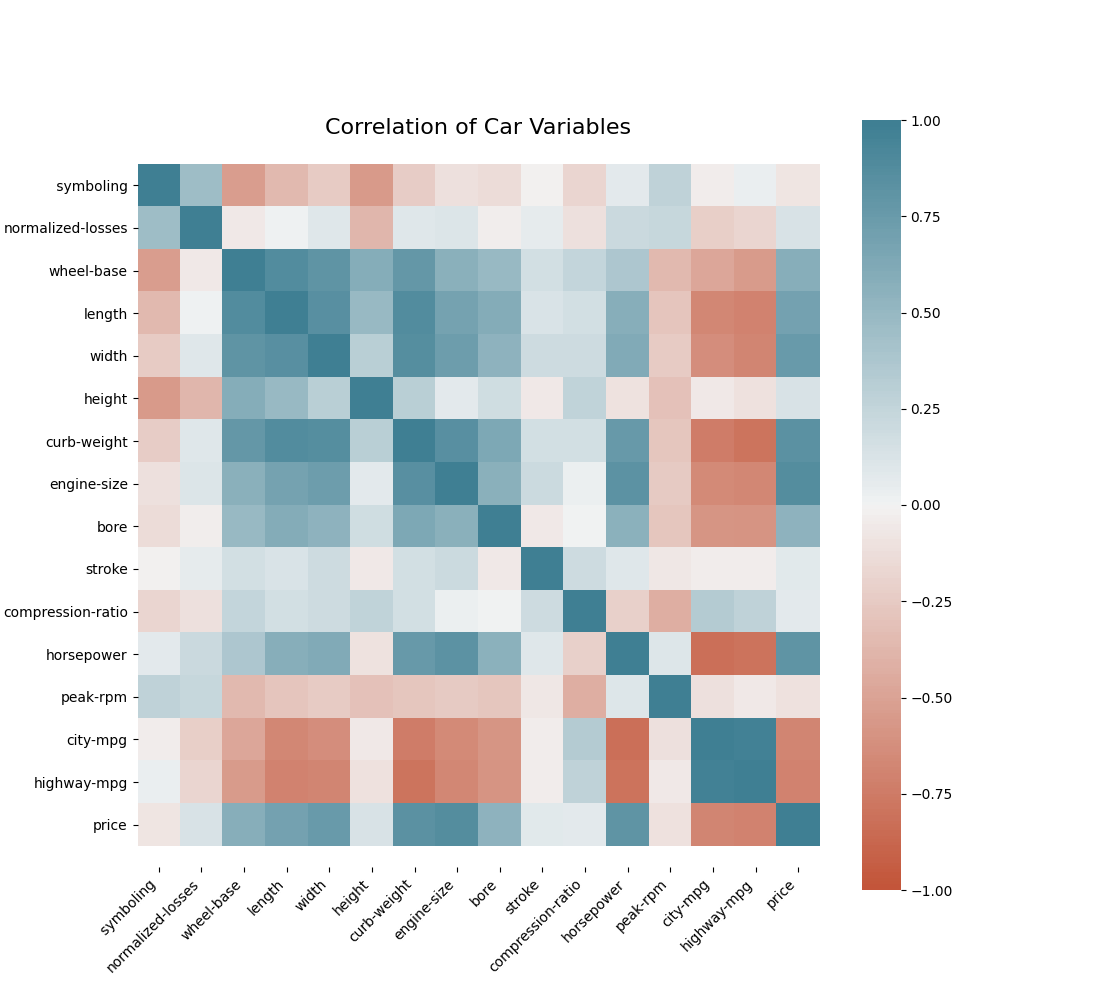
Time series data is often paired with multivariate linear regression models since the features that we want to forecast are numerical. The LSE and OLS models used in this experiment help us to forecast the price of an automobile given the independent variables: ‘normalized-losses’, ‘wheel-base’, ‘length’, ‘width’, ‘height’, ‘curb-weight’, ‘engine-size’, ‘bore’, ‘stroke’, ‘compression-ratio’, ‘horsepower’, ‘peak-rpm’, ‘city-mpg’ and ‘highway-mpg’.

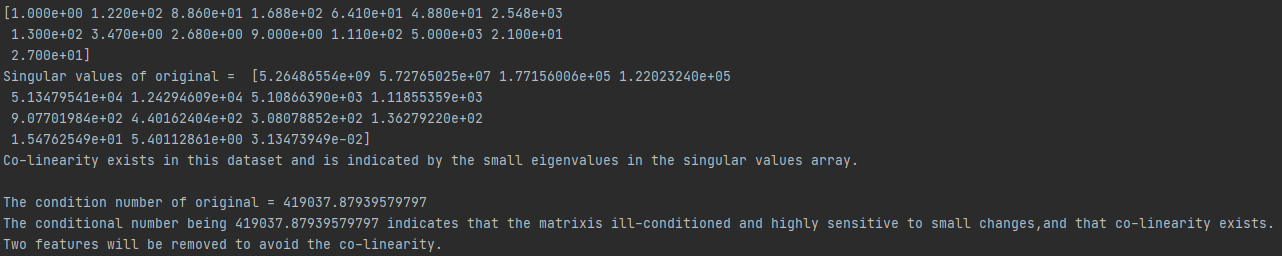
Multicollinearity is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model. To check for multicollinearity, a heatmap was created using the independent variables using the seaborn package, SVD analysis was conducted, and backward stepwise regression was used to determine which variables can be dropped to remove multicollinearity and improve the accuracy of the forecast.

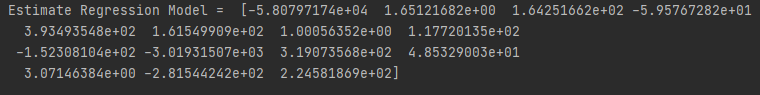
The formula for the LSE model in order to find the unknown values of beta is:

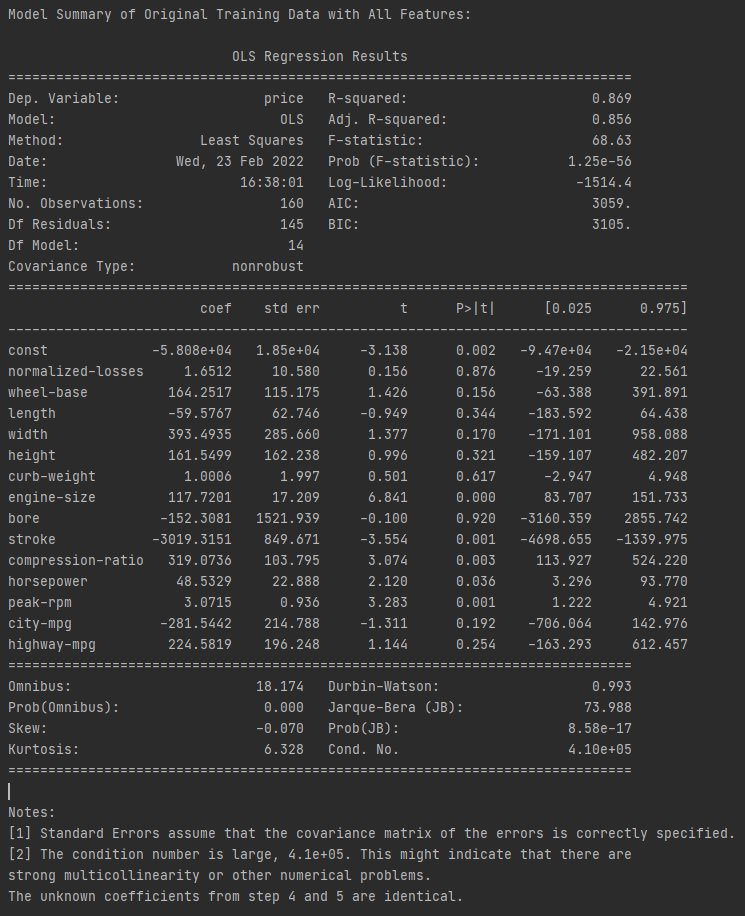


**Answers to Lab Questions:**

**2. **

**3.** 

**4.** 

**5.** 

**6. Summary of Training Data After Removing "bore" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.869**

**Model: OLS Adj. R-squared: 0.857**

**Method: Least Squares F-statistic: 74.41**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 1.43e-57**

**Time: 16:38:01 Log-Likelihood: -1514.4**

**No. Observations: 160 AIC: 3057.**

**Df Residuals: 146 BIC: 3100.**

**Df Model: 13**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -5.88e+04 1.7e+04 -3.457 0.001 -9.24e+04 -2.52e+04**

**normalized-losses 1.7897 10.453 0.171 0.864 -18.869 22.449**

**wheel-base 163.3571 114.438 1.427 0.156 -62.812 389.526**

**length -60.2266 62.197 -0.968 0.334 -183.150 62.697**

**width 394.0880 284.628 1.385 0.168 -168.435 956.611**

**height 163.2132 160.836 1.015 0.312 -154.655 481.082**

**curb-weight 1.0186 1.982 0.514 0.608 -2.899 4.937**

**engine-size 117.8979 17.059 6.911 0.000 84.183 151.612**

**stroke -3005.5437 835.606 -3.597 0.000 -4656.990 -1354.098**

**compression-ratio 317.1114 101.580 3.122 0.002 116.354 517.869**

**horsepower 47.9373 22.026 2.176 0.031 4.407 91.467**

**peak-rpm 3.1064 0.865 3.590 0.000 1.396 4.817**

**city-mpg -279.5013 213.090 -1.312 0.192 -700.640 141.638**

**highway-mpg 224.7105 195.577 1.149 0.252 -161.817 611.238**

**==============================================================================**

**Omnibus: 17.965 Durbin-Watson: 0.990**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 72.315**

**Skew: -0.064 Prob(JB): 1.98e-16**

**Kurtosis: 6.291 Cond. No. 3.78e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 3.78e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "normalized-losses" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.869**

**Model: OLS Adj. R-squared: 0.858**

**Method: Least Squares F-statistic: 81.15**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 1.57e-58**

**Time: 16:38:01 Log-Likelihood: -1514.4**

**No. Observations: 160 AIC: 3055.**

**Df Residuals: 147 BIC: 3095.**

**Df Model: 12**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -5.871e+04 1.69e+04 -3.465 0.001 -9.22e+04 -2.52e+04**

**wheel-base 166.0377 112.987 1.470 0.144 -57.251 389.327**

**length -60.6161 61.950 -0.978 0.329 -183.044 61.812**

**width 398.8909 282.306 1.413 0.160 -159.011 956.792**

**height 153.9118 150.883 1.020 0.309 -144.269 452.092**

**curb-weight 1.0483 1.968 0.533 0.595 -2.842 4.938**

**engine-size 117.5226 16.862 6.970 0.000 84.200 150.845**

**stroke -3007.6949 832.748 -3.612 0.000 -4653.400 -1361.990**

**compression-ratio 316.9487 101.240 3.131 0.002 116.876 517.022**

**horsepower 48.0568 21.942 2.190 0.030 4.695 91.419**

**peak-rpm 3.1160 0.861 3.621 0.000 1.415 4.817**

**city-mpg -286.6455 208.273 -1.376 0.171 -698.242 124.951**

**highway-mpg 231.9652 190.300 1.219 0.225 -144.113 608.043**

**==============================================================================**

**Omnibus: 17.991 Durbin-Watson: 0.994**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 72.485**

**Skew: -0.066 Prob(JB): 1.82e-16**

**Kurtosis: 6.295 Cond. No. 3.77e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 3.77e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "curb-weight" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.869**

**Model: OLS Adj. R-squared: 0.859**

**Method: Least Squares F-statistic: 88.93**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 1.87e-59**

**Time: 16:38:01 Log-Likelihood: -1514.6**

**No. Observations: 160 AIC: 3053.**

**Df Residuals: 148 BIC: 3090.**

**Df Model: 11**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -6.006e+04 1.67e+04 -3.594 0.000 -9.31e+04 -2.7e+04**

**wheel-base 178.8338 110.135 1.624 0.107 -38.807 396.475**

**length -53.8332 60.480 -0.890 0.375 -173.348 65.682**

**width 425.3626 277.222 1.534 0.127 -122.461 973.186**

**height 158.2753 150.296 1.053 0.294 -138.727 455.278**

**engine-size 119.7368 16.301 7.345 0.000 87.523 151.950**

**stroke -2973.4431 828.249 -3.590 0.000 -4610.164 -1336.722**

**compression-ratio 337.4087 93.440 3.611 0.000 152.760 522.057**

**horsepower 51.3240 21.016 2.442 0.016 9.795 92.853**

**peak-rpm 2.9923 0.827 3.620 0.000 1.359 4.626**

**city-mpg -285.6199 207.760 -1.375 0.171 -696.179 124.939**

**highway-mpg 208.6321 184.739 1.129 0.261 -156.436 573.700**

**==============================================================================**

**Omnibus: 17.907 Durbin-Watson: 1.021**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 72.325**

**Skew: -0.043 Prob(JB): 1.97e-16**

**Kurtosis: 6.293 Cond. No. 3.35e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 3.35e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "length" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.868**

**Model: OLS Adj. R-squared: 0.859**

**Method: Least Squares F-statistic: 97.88**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 2.72e-60**

**Time: 16:38:01 Log-Likelihood: -1515.0**

**No. Observations: 160 AIC: 3052.**

**Df Residuals: 149 BIC: 3086.**

**Df Model: 10**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -5.875e+04 1.66e+04 -3.531 0.001 -9.16e+04 -2.59e+04**

**wheel-base 133.0737 97.332 1.367 0.174 -59.256 325.404**

**width 348.9836 263.423 1.325 0.187 -171.544 869.511**

**height 120.7793 144.171 0.838 0.404 -164.104 405.662**

**engine-size 118.9571 16.266 7.313 0.000 86.814 151.100**

**stroke -2943.7248 826.999 -3.560 0.000 -4577.885 -1309.564**

**compression-ratio 328.8318 92.877 3.541 0.001 145.306 512.358**

**horsepower 50.6306 20.986 2.413 0.017 9.161 92.100**

**peak-rpm 3.0961 0.818 3.786 0.000 1.480 4.712**

**city-mpg -218.9137 193.637 -1.131 0.260 -601.544 163.716**

**highway-mpg 171.0471 179.724 0.952 0.343 -184.089 526.183**

**==============================================================================**

**Omnibus: 17.374 Durbin-Watson: 1.006**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 66.791**

**Skew: -0.078 Prob(JB): 3.14e-15**

**Kurtosis: 6.161 Cond. No. 3.34e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 3.34e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "height" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.867**

**Model: OLS Adj. R-squared: 0.859**

**Method: Least Squares F-statistic: 108.9**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 3.58e-61**

**Time: 16:38:01 Log-Likelihood: -1515.4**

**No. Observations: 160 AIC: 3051.**

**Df Residuals: 150 BIC: 3082.**

**Df Model: 9**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -5.297e+04 1.51e+04 -3.502 0.001 -8.29e+04 -2.31e+04**

**wheel-base 178.8177 80.493 2.222 0.028 19.771 337.865**

**width 305.6295 258.033 1.184 0.238 -204.219 815.478**

**engine-size 117.4667 16.153 7.272 0.000 85.550 149.383**

**stroke -3048.7616 816.626 -3.733 0.000 -4662.338 -1435.185**

**compression-ratio 331.0819 92.746 3.570 0.000 147.825 514.339**

**horsepower 50.5417 20.965 2.411 0.017 9.116 91.967**

**peak-rpm 3.0132 0.811 3.715 0.000 1.411 4.616**

**city-mpg -215.8108 193.409 -1.116 0.266 -597.970 166.348**

**highway-mpg 168.0934 179.510 0.936 0.351 -186.602 522.789**

**==============================================================================**

**Omnibus: 17.709 Durbin-Watson: 1.012**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 70.213**

**Skew: -0.059 Prob(JB): 5.67e-16**

**Kurtosis: 6.243 Cond. No. 3.04e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 3.04e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "highway-mpg" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.866**

**Model: OLS Adj. R-squared: 0.859**

**Method: Least Squares F-statistic: 122.5**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 4.83e-62**

**Time: 16:38:01 Log-Likelihood: -1515.9**

**No. Observations: 160 AIC: 3050.**

**Df Residuals: 151 BIC: 3077.**

**Df Model: 8**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -4.952e+04 1.47e+04 -3.377 0.001 -7.85e+04 -2.06e+04**

**wheel-base 168.3419 79.679 2.113 0.036 10.911 325.772**

**width 281.3419 256.621 1.096 0.275 -225.690 788.374**

**engine-size 115.4832 16.007 7.215 0.000 83.857 147.109**

**stroke -2979.3949 812.928 -3.665 0.000 -4585.577 -1373.213**

**compression-ratio 328.8373 92.677 3.548 0.001 145.726 511.948**

**horsepower 52.5411 20.848 2.520 0.013 11.350 93.732**

**peak-rpm 3.0272 0.811 3.735 0.000 1.426 4.629**

**city-mpg -52.2924 83.114 -0.629 0.530 -216.508 111.923**

**==============================================================================**

**Omnibus: 17.933 Durbin-Watson: 0.987**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 72.250**

**Skew: -0.056 Prob(JB): 2.05e-16**

**Kurtosis: 6.290 Cond. No. 2.94e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 2.94e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "city-mpg" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.866**

**Model: OLS Adj. R-squared: 0.860**

**Method: Least Squares F-statistic: 140.5**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 4.77e-63**

**Time: 16:38:01 Log-Likelihood: -1516.1**

**No. Observations: 160 AIC: 3048.**

**Df Residuals: 152 BIC: 3073.**

**Df Model: 7**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -5.448e+04 1.23e+04 -4.414 0.000 -7.89e+04 -3.01e+04**

**wheel-base 183.9395 75.574 2.434 0.016 34.628 333.251**

**width 314.8122 250.547 1.256 0.211 -180.193 809.817**

**engine-size 112.2363 15.122 7.422 0.000 82.360 142.113**

**stroke -2984.4499 811.271 -3.679 0.000 -4587.273 -1381.627**

**compression-ratio 305.5303 84.783 3.604 0.000 138.026 473.035**

**horsepower 60.6620 16.339 3.713 0.000 28.381 92.943**

**peak-rpm 2.9749 0.805 3.697 0.000 1.385 4.565**

**==============================================================================**

**Omnibus: 17.915 Durbin-Watson: 0.991**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 71.425**

**Skew: -0.078 Prob(JB): 3.09e-16**

**Kurtosis: 6.269 Cond. No. 2.48e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 2.48e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "width" Feature:**

**OLS Regression Results**

**==============================================================================**

**Dep. Variable: price R-squared: 0.865**

**Model: OLS Adj. R-squared: 0.859**

**Method: Least Squares F-statistic: 163.0**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 7.81e-64**

**Time: 16:38:01 Log-Likelihood: -1516.9**

**No. Observations: 160 AIC: 3048.**

**Df Residuals: 153 BIC: 3069.**

**Df Model: 6**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**const -4.164e+04 6932.165 -6.006 0.000 -5.53e+04 -2.79e+04**

**wheel-base 250.5779 53.941 4.645 0.000 144.012 357.144**

**engine-size 113.9081 15.092 7.548 0.000 84.093 143.723**

**stroke -2945.3206 812.205 -3.626 0.000 -4549.905 -1340.736**

**compression-ratio 321.4083 83.994 3.827 0.000 155.471 487.346**

**horsepower 66.9018 15.595 4.290 0.000 36.091 97.712**

**peak-rpm 3.0111 0.806 3.737 0.000 1.419 4.603**

**==============================================================================**

**Omnibus: 18.109 Durbin-Watson: 1.032**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 70.826**

**Skew: -0.128 Prob(JB): 4.17e-16**

**Kurtosis: 6.249 Cond. No. 1.39e+05**

**==============================================================================**

**Notes:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[2] The condition number is large, 1.39e+05. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "const" Feature:**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.950**

**Model: OLS Adj. R-squared (uncentered): 0.948**

**Method: Least Squares F-statistic: 491.9**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 9.78e-98**

**Time: 16:38:01 Log-Likelihood: -1533.8**

**No. Observations: 160 AIC: 3080.**

**Df Residuals: 154 BIC: 3098.**

**Df Model: 6**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**wheel-base 11.5261 40.344 0.286 0.775 -68.174 91.226**

**engine-size 109.6500 16.704 6.564 0.000 76.652 142.648**

**stroke -3644.9239 890.662 -4.092 0.000 -5404.416 -1885.432**

**compression-ratio 257.0719 92.310 2.785 0.006 74.715 439.428**

**horsepower 83.0938 17.020 4.882 0.000 49.470 116.717**

**peak-rpm -0.1730 0.672 -0.257 0.797 -1.501 1.155**

**==============================================================================**

**Omnibus: 15.118 Durbin-Watson: 0.891**

**Prob(Omnibus): 0.001 Jarque-Bera (JB): 50.110**

**Skew: -0.063 Prob(JB): 1.31e-11**

**Kurtosis: 5.739 Cond. No. 1.61e+04**

**==============================================================================**

**Notes:**

**[1] R² is computed without centering (uncentered) since the model does not contain a constant.**

**[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[3] The condition number is large, 1.61e+04. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "wheel-base" Feature:**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.950**

**Model: OLS Adj. R-squared (uncentered): 0.949**

**Method: Least Squares F-statistic: 593.8**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 3.97e-99**

**Time: 16:38:01 Log-Likelihood: -1533.9**

**No. Observations: 160 AIC: 3078.**

**Df Residuals: 155 BIC: 3093.**

**Df Model: 5**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**engine-size 112.2083 14.059 7.981 0.000 84.437 139.980**

**stroke -3590.0499 867.124 -4.140 0.000 -5302.956 -1877.144**

**compression-ratio 267.1782 85.010 3.143 0.002 99.251 435.105**

**horsepower 81.4598 15.983 5.097 0.000 49.887 113.033**

**peak-rpm -0.0375 0.475 -0.079 0.937 -0.976 0.901**

**==============================================================================**

**Omnibus: 15.680 Durbin-Watson: 0.897**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 53.491**

**Skew: -0.084 Prob(JB): 2.42e-12**

**Kurtosis: 5.828 Cond. No. 1.58e+04**

**==============================================================================**

**Notes:**

**[1] R² is computed without centering (uncentered) since the model does not contain a constant.**

**[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**[3] The condition number is large, 1.58e+04. This might indicate that there are**

**strong multicollinearity or other numerical problems.**

**Summary of Training Data After Removing "peak-rpm" Feature:**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.950**

**Model: OLS Adj. R-squared (uncentered): 0.949**

**Method: Least Squares F-statistic: 747.0**

**Date: Wed, 23 Feb 2022 Prob (F-statistic): 1.37e-100**

**Time: 16:38:01 Log-Likelihood: -1533.9**

**No. Observations: 160 AIC: 3076.**

**Df Residuals: 156 BIC: 3088.**

**Df Model: 4**

**Covariance Type: nonrobust**

**=====================================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------------**

**engine-size 112.7274 12.386 9.101 0.000 88.262 137.193**

**stroke -3651.9222 369.446 -9.885 0.000 -4381.684 -2922.161**

**compression-ratio 267.6170 84.557 3.165 0.002 100.593 434.641**

**horsepower 80.8756 14.121 5.727 0.000 52.983 108.768**

**==============================================================================**

**Omnibus: 15.796 Durbin-Watson: 0.896**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 54.335**

**Skew: -0.084 Prob(JB): 1.59e-12**

**Kurtosis: 5.850 Cond. No. 229.**

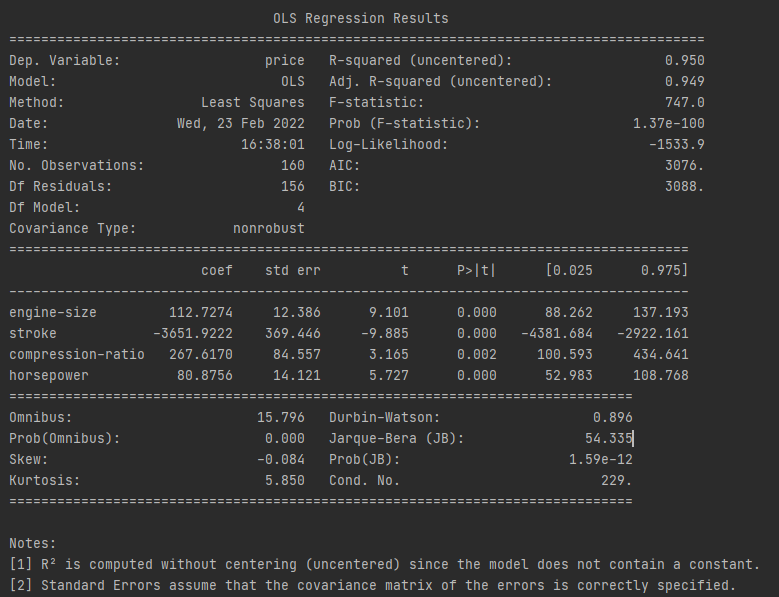
**==============================================================================**

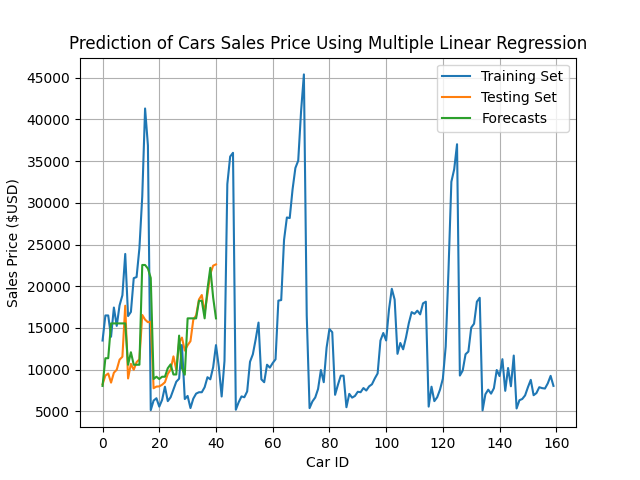
**Notes:**

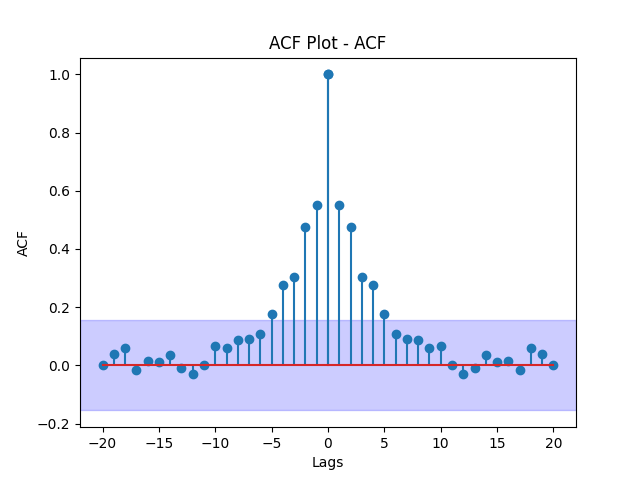
**[1] R² is computed without centering (uncentered) since the model does not contain a constant.**

**[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

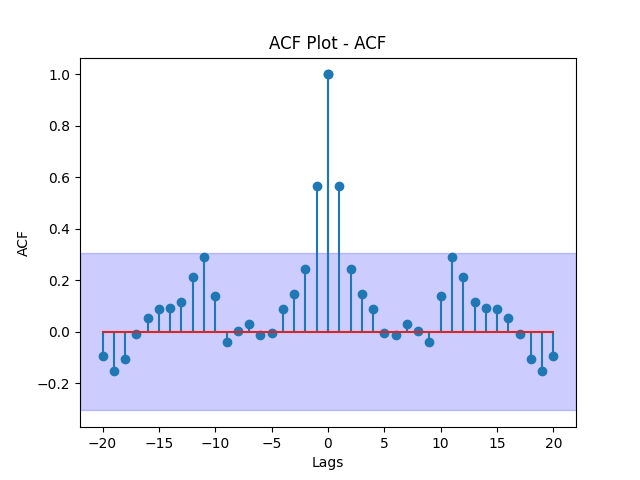
**The features recommended for keeping are engine-size, stroke, compression-ratio, and horsepower. The rest are recommended to be eliminated.**

**7.** 

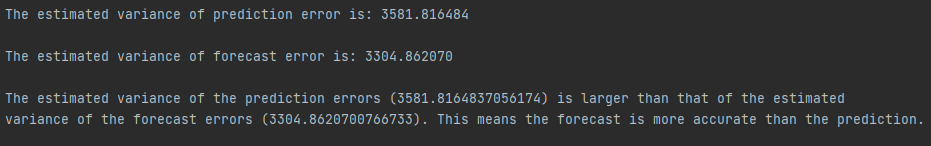
**8. **

**9. **

Since the ACF plots shows values falling within the insignificant zone after 5 lags, we assume the data is stationary**.**

**10. **

Since the ACF plots shows values falling within the insignificant zone after less than 5 lags, we assume the data is stationary**.**

**11.** 

**12.**

**T-test:** Since the t-values for the four independent variables in the final model are all greater than their respective p-values, we reject the null hypothesis and conclude that they are all significant.

**F-Test:** The p-values for each independent variable associated with the f-statistic are all < 0.05 which lets us assume that each independent variable is related to the dependent variable.

**Conclusion:**

The OLS and LSE models provided the same values for the unknown coefficients indicating that both methods are useful in forecasting time series data. After using backward stepwise regression, we found that the four independent variables worth keeping are engine-size, stroke, compression-ratio, and horsepower. These independent variables are best at indicating the price of an automobile. Removing multicollinearity is key in making sure models forecast with proper accuracy, otherwise the results will be skewed.

**Appendix:**

import pandas as pd  
import numpy as np  
from numpy import linalg as la  
from sklearn.model\_selection import train\_test\_split  
import statsmodels.api as sm  
import seaborn as sns  
import matplotlib.pyplot as plt  
import warnings  
from scipy.stats import ttest\_ind  
warnings.filterwarnings('ignore')  
  
df = pd.read\_csv(r'C:\Users\brear\OneDrive\Desktop\Grad School\Time-Series-Analysis-and-Moldeing\Datasets\auto.clean.csv')  
#print(df.columns)  
#print(df.head())  
#print(df.shape)  
  
y = df['price'].copy()  
X = df[['normalized-losses', 'wheel-base', 'length', 'width',  
'height', 'curb-weight', 'engine-size', 'bore', 'stroke', 'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg']]  
X = sm.add\_constant(X, prepend=True)  
  
  
# question 1  
# train test split (80/20)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, shuffle=False, test\_size=0.2)  
X\_test.reset\_index(drop=True, inplace=True)  
y\_test.reset\_index(drop=True, inplace=True)  
  
# question 2  
# correlation heatmap  
corr = df.corr()  
fig = plt.figure(figsize=(11, 10))  
ax = sns.heatmap(corr, vmin=-1, vmax=1, center=0, cmap=sns.diverging\_palette(20, 220, n=200), square=True)  
bottom, top = ax.get\_ylim()  
ax.set\_ylim(bottom + 0.5, top - 0.5)  
ax.set\_xticklabels(ax.get\_xticklabels(), rotation=45, horizontalalignment='right')  
plt.title('Correlation of Car Variables', fontsize=16)  
plt.show()  
  
# question 3  
# SVD analysis  
X\_matrix = X\_train.values  
print(X\_matrix[0])  
y\_matrix = y\_train.values  
H = np.matmul(X\_matrix.T, X\_matrix)  
  
u, s, v = np.linalg.svd(H) #, full\_matrices=True)  
print('Singular values of original = ', s)  
print('Co-linearity exists in this dataset and is indicated by the small eigenvalues '  
 'in the singular values array.')  
print('\nThe condition number of original = {}'.format(la.cond(X)),  
 '\nThe conditional number being {} indicates that the matrix'  
 'is ill-conditioned and highly sensitive to small changes,'  
 'and that co-linearity exists.'.format(la.cond(X)))  
print('Two features will be removed to avoid the co-linearity.')  
  
# question 4  
# estimate the regression model using LSE method  
estimate\_model = np.matmul(np.linalg.inv(np.matmul(X\_matrix.T, X\_matrix)), np.matmul(X\_matrix.T, y\_matrix))  
print('Estimate Regression Model = ', estimate\_model)  
print()  
  
# question 5  
#Use OLS function to find the unknown coefficients  
model = sm.OLS(y\_train, X\_train).fit()  
print('Model Summary of Original Training Data with All Features: \n')  
print(model.summary())  
print('The unknown coefficients from step 4 and 5 are identical.')  
  
# question 6  
#Use backward stepwise regression to reduce the feature space dimension  
  
#----------------------  
#Removing 'bore' feature  
#----------------------  
X\_train.drop(['bore'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "bore" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'normalized-losses' feature  
#----------------------  
X\_train.drop(['normalized-losses'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "normalized-losses" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'curb-weight' feature  
#----------------------  
X\_train.drop(['curb-weight'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "curb-weight" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'length' feature  
#----------------------  
X\_train.drop(['length'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "length" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'height' feature  
#----------------------  
X\_train.drop(['height'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "height" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'highway-mpg' feature  
#----------------------  
X\_train.drop(['highway-mpg'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "highway-mpg" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'city-mpg' feature  
#----------------------  
X\_train.drop(['city-mpg'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "city-mpg" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'width' feature  
#----------------------  
X\_train.drop(['width'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "width" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'const' feature  
#----------------------  
X\_train.drop(['const'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "const" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'wheel-base' feature  
#----------------------  
X\_train.drop(['wheel-base'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "wheel-base" Feature:\n')  
print(model.summary())  
  
#----------------------  
#Removing 'peak-rpm' feature  
#----------------------  
X\_train.drop(['peak-rpm'], axis=1, inplace=True)  
model = sm.OLS(y\_train, X\_train).fit()  
print('\nSummary of Training Data After Removing "peak-rpm" Feature:\n')  
print(model.summary())  
  
print('The features recommended for keeping are engine-size, stroke, compression-ratio, and horsepower.'  
 'The rest are recommended to be eliminated.')  
  
# question 7  
#Use OLS function on the reduced feature space  
# model output from above  
  
# question 8  
#drop the columns in X\_test that were dropped in X\_train  
X\_test.drop(['const', 'normalized-losses', 'wheel-base', 'length', 'width', 'height', 'curb-weight', 'bore',  
 'peak-rpm', 'city-mpg', 'highway-mpg'], axis=1, inplace=True)  
  
#prediction values for the prediction (train) set and forecast (test) set  
predictions\_pred = model.predict(X\_train)  
predictions\_fore = model.predict(X\_test)  
  
#plot the training set, testing set, and forecasts of regression model  
plt.figure()  
plt.plot(y\_train, label='Training Set')  
plt.plot(y\_test, label='Testing Set')  
plt.plot(predictions\_fore, label='Forecasts')  
plt.xlabel('Car ID')  
plt.ylabel('Sales Price ($USD)')  
plt.title('Prediction of Cars Sales Price Using Multiple Linear Regression')  
plt.grid()  
plt.legend()  
plt.show()  
  
# question 9 & 10  
  
#calculate the predictions error  
df\_pred = pd.DataFrame([y\_train, predictions\_pred]).transpose()  
df\_pred.columns = ['Y\_train', 'Predictions']  
df\_pred['pred\_error'] = df\_pred['Y\_train'] - df\_pred['Predictions']  
  
#calculate the forecast error  
df\_fore = pd.DataFrame([y\_test, predictions\_fore]).transpose()  
df\_fore.columns = ['Y\_test', 'Forecast']  
df\_fore['forecast\_error'] = df\_fore['Y\_test'] - df\_fore['Forecast']  
  
def ACF(timeseries\_data, lags, metric=''):  
 auto\_corr = []  
 timeseries\_data\_mean = np.mean(timeseries\_data)  
 length = len(timeseries\_data)  
 denominator = 0 # 0th lag adjusted  
 x\_axis = np.arange(0, lags+1)  
 m = 1.96/np.sqrt(length)  
  
 for denom\_t in range(0, length):  
 denominator = denominator + (timeseries\_data[denom\_t] - timeseries\_data\_mean) \*\* 2  
  
 for tau in range(0, lags+1):  
 numerator = 0  
 for num\_t in range(tau, length):  
 numerator = numerator + (timeseries\_data[num\_t] - timeseries\_data\_mean) \* (  
 timeseries\_data[num\_t - tau] - timeseries\_data\_mean)  
 auto\_corr.append((numerator / denominator))  
  
 plt.stem(x\_axis, auto\_corr, use\_line\_collection=True)  
 plt.stem(-1 \* x\_axis, auto\_corr, use\_line\_collection=True)  
 plt.title(f"ACF Plot - {metric}")  
 plt.xlabel("Lags")  
 plt.ylabel("ACF")  
 plt.axhspan(-m, m, alpha=0.2, color='blue')  
 # use plt.show() in main for graphs to show  
 return auto\_corr  
  
#Find ACF of prediction error and forecast error  
ACF(df\_pred['pred\_error'].to\_numpy(), 20, 'ACF')  
plt.show()  
ACF(df\_fore['forecast\_error'].to\_numpy(), 20, 'ACF')  
plt.show()  
  
# question 11  
  
#calculate estimated variance for prediction errors  
sse\_pred = 0  
for i in range(len(df\_pred)):  
 sse\_pred += (df\_pred.iloc[i, 2]) \*\* 2  
  
variance\_pred = np.sqrt(sse\_pred / (len(df\_pred) - 4 - 1))  
print('\nThe estimated variance of prediction error is: %0.6f' % variance\_pred)  
  
#calculate estimated variance for forecast errors  
sse\_fore = 0  
for i in range(len(df\_fore)):  
 sse\_fore += (df\_fore.iloc[i, 2]) \*\* 2  
  
variance\_fore = np.sqrt(sse\_fore / (len(df\_fore) - 1 - 1))  
print('\nThe estimated variance of forecast error is: %0.6f' % variance\_fore)  
  
# results explained  
print('\nThe estimated variance of the prediction errors ({}) is larger \  
than that of the estimated\nvariance of the forecast errors ({}). This \  
means the forecast is more accurate than the prediction.'.format(variance\_pred, variance\_fore))