

## Technical Programming II

### Practical 10: Simple Linear Regression

#### Tasks

**The data file for Questions 1 through 9 is available from the Seaborn website. To load this data into a DataFrame, run this code:**

```
data = sns.load_dataset('diamonds')
```

1. Use a condensed heatmap to identify correlations for the price column. Sort the results, include annotations that format the values with three decimal places, and remove the color bar.
2. Use the column with the strongest correlation to split the data into test and training datasets, where the test dataset consists of 30% of the total dataset. Be sure to specify a value for the `random_state` parameter.
3. Create a `LinearRegression` object, and then fit the training dataset to the model.
4. Score the model using the test dataset.
5. Predict the `y` values based on the `x` values in the test dataset, and store the results in a variable.
6. Put the predicted values in a DataFrame.
7. Join the `y_test` and predicted data with the `x_test` data, save the joined data in a new DataFrame, and then display the first five rows of data.
8. Melt the actual and predicted price columns together, assigning appropriate values to the variable and value names.
9. Use a Seaborn line plot without a confidence interval to plot the predicted data.

**The data file for the next questions is available from the Seaborn website. To load this data into a DataFrame, run this code:**

```
mpg = sns.load_dataset('mpg')
```

10. Get the correlation data for the mpg column. Sort the results to make it easier to see the columns with the stronger correlations.
11. Use the column with the strongest positive or negative correlation to the mpg column to split the data into test and training datasets, where the test dataset consists of 20% of the total dataset. Be sure to specify a value for the `random_state` parameter.

12. Create a linear regression model from the training dataset.
13. Score the model using the test dataset.
14. Score the model using the training dataset.
15. Predict the y values based on the x values in the test dataset, and store the results in a variable. Then, put the predicted values in a DataFrame.
16. Join the y\_test and predicted data with the x\_test data, save the combined data in a new DataFrame, and then display the first five rows of data.
17. Add the residuals to the new DataFrame.
18. Plot the residuals in a Seaborn KDE plot.
19. Melt the actual and predicted MPG columns together, assigning appropriate values to the variable and value names.
20. Use a Seaborn scatter plot to plot the predicted data.

-END-