Assignment

explore the available data, create visualisations to identify trends, and extract meaningful insights to inform decision-making.

a PDF report of 1,000 words (+/- 20%) indicating your approach, **thought process, results, conclusions, and any recommendations** you would like to make (as a PDF file)

Explain everything everywhere.

Context

Analytical approachand thought process:

Dependent variable

Positively skewed distribution, in line with the nature of loyalty points. The size of dataset should still allow meaningful regression model

Independent variables

Age not linear

Spending score is approximately linear

Based on the metadata this is the same score like loyalty - be careful

Renumeration is approximately linear but there is a heteroskedastic error

I tried MLR but ended up with a p value below 0.01

I must transform the data and see what it can show with spending score

Results

Conclusion

Recommendation

**Assignment3**

**Approach from MLR earlier**

1. **Open** a new Jupyter Notebook (Python3), and import the relevant modules and packages (e.g. Pandas, NumPy, statsmodels.api, sklearn, Matplotlib).
2. **Import** the ecommerce\_data.csv file (see the **Data** folder in the ZIP file you downloaded from [1.0 Focus of the week](https://fourthrev.instructure.com/courses/464/pages/1-dot-0-focus-of-the-week)), and **view** it as a Pandas DataFrame to check if the data is valid. (**Option**: Use (df.head() and df.info() to check the information against the table above.)
3. **Familiarise** yourself with the metadata (ecommerce\_metadata.txt).
4. **Define**the independent and dependent variables for your regression model. (**Hint:** We require more than one independent variable in MLR.)
   1. Specify and fit the model to use.
   2. Call the predictions for the independent variable.
   3. Check the value of the R2, intercept, and coefficients.
   4. Make predictions (e.g. 5.75 and 15.2).
5. **Train and test**the MLR.
   1. Dividethe data set into training and test sets. Specify the random\_state as 42.
   2. Train the model with the statsmodel of the OLS library.
   3. Set the predicted response vector.
   4. Call and print a summary of the model.
   5. Print the multi.score of the model based on the train data set.
6. **Run**a regression test.
   1. Fit the model with the fit() method.
   2. Call and print the predictions for X in the test set.
   3. Print the R2 value.
7. **Check** for multicollinearity.
   1. Create a variable to store the x\_train data set. (**Hint:** Leverage the sm.add\_constant() function.)
   2. Create an empty DataFrame.
   3. Calculate and print the VIF values.
   4. Determine whether heteroscedasticity is present. (**Hint:** Leverage the sms.het\_breuschpagan() function.)
8. **Evaluate** the model by determining the mean absolute error and mean square error.
9. **Summarise**the results in a sentence or two in your Notebook.