Programming & Maths for AI – Task 2

Emre Dogan, Hui Zheng, Éamonn Ó Cearnaigh

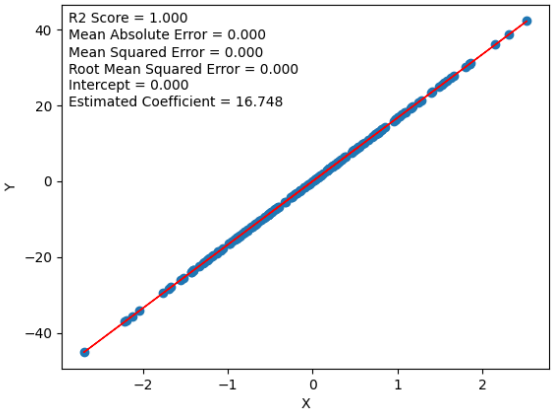
https://github.com/EamonnOCearnaigh/LinearRegression

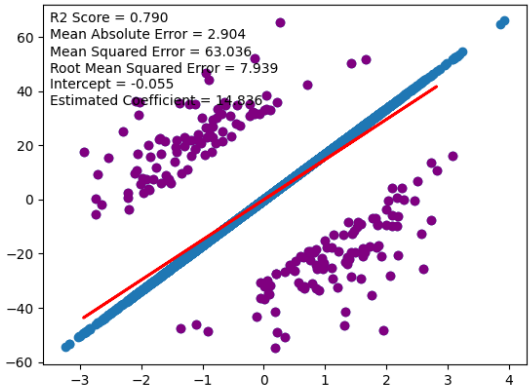
The two factors selected for this Question are: (1) Outliers and (2) Multicollinearity.

Outliers

To determine the **number of outliers** and the **degree to which they deviate from the mean or standard deviation,** the following needs to be performed: generate a **clean dataset** with a **mean of almost 0** and a **standard deviation of 1**. This will act as our **baseline**, and we will slowly adjust the variables mentioned above to see how they impact the performance of the model.

The model in all cases will be trained using Sklearn's **LinearRegression** Model with the default parameters. The data used to train the model will be split using a standard 80-20 train-test split. The number of samples used for the clean dataset is 5000. An example of what outliers added to this clean dataset looks like, is shown below:





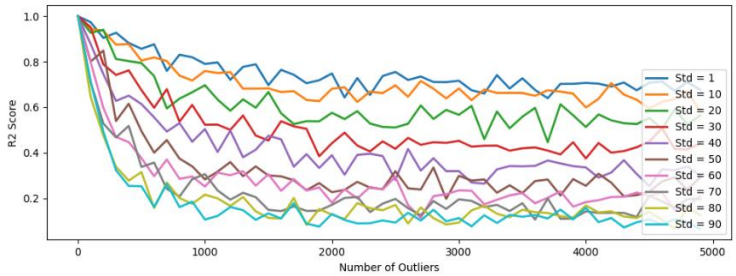
The image above was generated by adding 1000 outliers in the outer 1% bounds of the data, with a standard deviation of 2.0) to the clean dataset. Note: The type of outliers used here have high leverage and high residual to exaggerate the effects outliers could have and to better see the trends in the data.

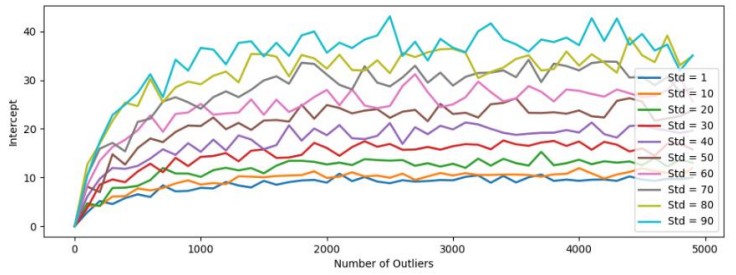
After running all these tests (from std of 1-90 in intervals of 10 std and number of outliers from 0-5000 in intervals of 100 totalling 500 iterations), it can be seen the outliers have a very negative impact on a Linear Regression model. Notably that Linear Regression is very sensitive to outliers. Linear correlation can be seen with every evaluation metric. One interesting note is that **most of the error from outliers is introduced from roughly the first 0-1500 outliers**, the number of outliers above this amount **contributes little to the overall error** in the regression model. Another thing that can be inferred from the "Estimated Coefficient" plot (which is the last plot) is that outliers have a big impact on both the **mean** and the **variance/standard deviation** of the data.

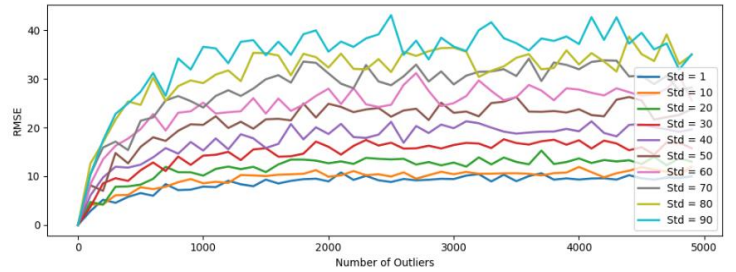
From an understanding of the theory, the results align with what would be expected from the presence of outliers in a linear regression dataset. Depending on the type of outlier, it can **increase error variance** and **bias estimates** as they could potentially have a lot of **influence** on the data. This will result in the model trying to fit a regression line through all the points the best it can, but because those outliers are **irrespective of most of the data** (typically less than **2-3 standard deviations from the mean**), and have a lot of influence, it will lead to an **unrepresentative model** which will fail to accurately predict the data.

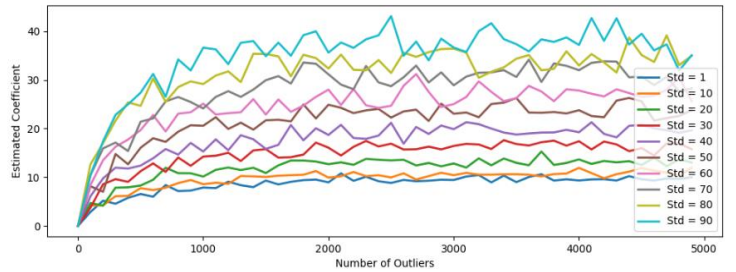
However, another thing to point out here is that **for small amounts of outliers, the error is not too great**. This is important because it might not always be the best idea to remove all outliers as they may contain **valuable information** that will be **lost otherwise** (for example, if looking at basketball

player statistics, one would not remove Michael Jordan or Kobe Bryant just because they are outliers).









So, an important skill to learn would be to identify **when** to remove outliers or not in addition to **how** to remove outliers. Unfortunately, the former skill depends on the dataset you are working with and the **context** that comes along with that dataset. But a general rule of thumb is to try and remove the outlier to see what impact it has on the model, whether a good or bad one. If one decides one cannot remove the outlier, but it is having a negative impact on the model, one could attempt to **transform** the data using either **square root** or **logarithmic** transformations to pull in high numbers in order to reduce their **leverage and residual**. One could also **impute** the outliers if one decides that the outliers are artificial. Another solution could be to just try another model that might fit the data better (maybe a **polynomial regression model** instead of linear?) or one that is less sensitive to outliers.

If it was decided that it might be best to remove the outliers, but there are too many to remove visually or by hand (which could be the case in real-life scenarios if the dataset is very noisy) as in our example, some of the following options could be explored:

If the data is univariate, then a simple box plot can be used to remove outliers visually or mathematically by removing all data roughly outside 1.5 times the inter-quartile range (**Tukey's box plot method**). Another method is to compute the **Internally Standardized Residuals/Z-score** of the data and reject values above a certain threshold. Z-score is a measure of how many standard deviations the data point is away from its mean, so a threshold of around 2-3 could work for most datasets. However, the issue with Z-score is that it is dependent on statistics such as standard deviation and mean which are highly susceptible to outliers, so one could use a more robust method called the **Median Absolute Deviation method** (MAD) which replaces the standard deviation and mean for things like median and median absolute deviation.

If the data is bivariate (as in our case) or multivariate, then a more sophisticated approach must be taken. One such approach would be a distance metric such as the **Mahalanobis Distance** (MD). The MD determines the distance between a data point and a distribution using their mean and covariance. It can be thought of as a multivariate generalization of z-score for univariate data. Like the Z-score, the MD also compares distance observations with a threshold and rejects data points above or below that threshold depending on how it is configured by the developer. However, much

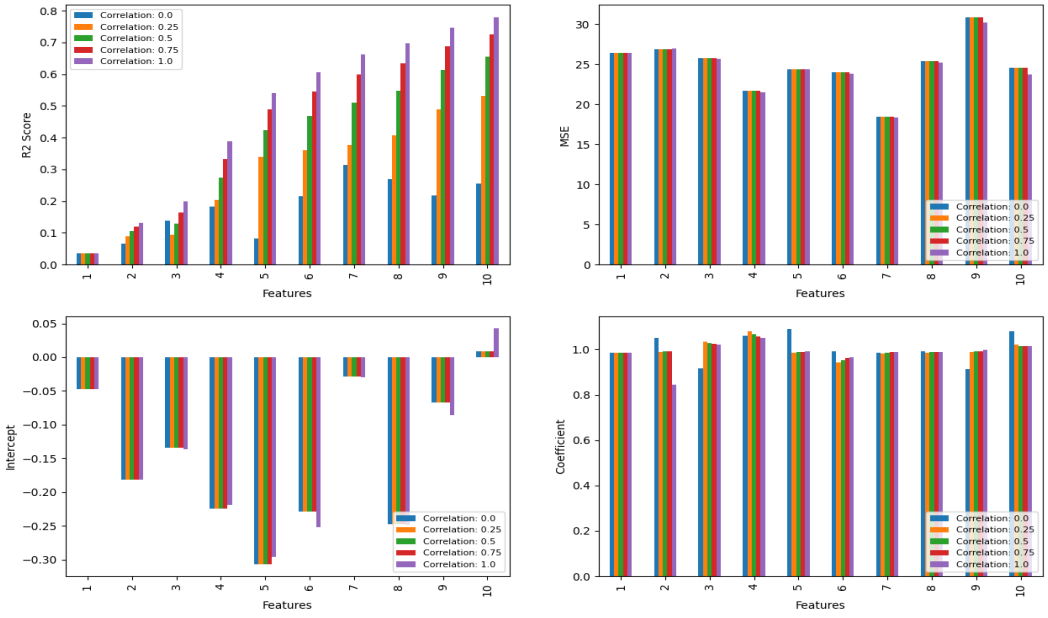
like with Z-score, it is very susceptible to outliers and so a more robust variation to the MD exists called the **Minimum Covariance Determinant** (MCD). The MCD uses a subset of the sample where the determinant of the covariance matrix is as small as possible. As opposed to MD, **Euclidean distance** can also be used to identify outliers, however, because it does not use the covariance matrix of the variables of the dataset, it cannot detect outliers based on the distribution pattern of the data meaning it might mark some non-outlier points as outliers. **Note that MD and Z-score assume the data has an underlying normal distribution.**

Apart from these two popular methods, others exist such as **k-nearest neighbours**, **DBSCAN**, and **isolation forests**. Although, these methods are more suited to removing outliers from classification problem sets which is out of the scope of the exploration.

Multi-Collinearity

For multi-collinearity, we will be generating a dataset with a varying number of features (from 1 to 10). Due to limited time and space in this report,we will be varying the collinearity between these features equally. This means if 5 features are present, all features will have the same collinearity between each other. Along with that, the dataset will have 5000 samples total with normal distribution noise of 5 standard deviations applied to it. All features were given the same actual coefficient of 1 when generating the dataset. The reason for the high std and low sample count is to exaggerate the effects of multi-collinearity in order to make it easier to analyse at a bigger scale.

The following graph are the findings after testing 1-10 features each with collinearity 0, 0.25, 0.5, 1.0. As can be seen, the higher the number of features, the higher the R2 score but also the higher the number of features the more variable the coefficient estimates are. The coefficient graph plots a coefficient accuracy where 1 is 100% accurate. These results are not completely what was expected. Based on the research conducted, multi-collinearity can lead to a wild fluctuation in coefficient estimates which is not exactly what is seen here. Even stranger, the R2 score is abysmal with only just 1 variable. We assume that this is an issue with the code that was used to generate and train the Linear Regression model. However, time constraints did not permit further investigation into this.



The research and academic study online suggest that, while multi-collinearity will cause wild variances in estimated coefficients, it might not necessarily affect the R2 score that much, and the

model might very likely still be able to accurately fit a Linear Regression model to a particular dataset. However, while this means is that one can accurately predict the response variable, one cannot reliably predict the effect of individual features on the response variable as these features aren't independent variables which will lead to issues when it comes to interpretability of the model. Another impact this can have is it could inflate the estimated coefficients, making it difficult to detect statistical significance. So, what can we do to fix this?

Firstly, to detect multi-collinearity, one can use what is called the Variance Inflation Factor (VIF). VIF calculates how much the variance of a coefficient is inflated because of its linear dependencies with other predictors. It allows the determination of the strength of the correlation between the various features in a dataset. When a variable has a VIF of 1, that means the tested feature is not correlated with the other feature. A VIF higher than 1 will mean it is in some way correlated with the other feature, the standard error is inflated and less likely that the coefficient will be statistically significant as mentioned before. An acceptable VIF threshold is actively debated online but as a general rule case anything below a VIF of 5-10 is considered to not be highly collinear

After detecting the highly collinear variables, one can remove the highly collinear variables as they are likely to provide redundant information. If the loss of information poses accuracy issues, another option would be to derive independent child features from the highly collinear ones and using those in the model. Another very popular method is to perform dimensionality reduction using methods such as Principal Component Analysis (PCA). PCA will aim to preserve the variances of the features while still representing them in fewer dimensions which can also help reveal dominant trends in the data.

Reflections

At the beginning of this project, the team members agreed on the work that each member needed to do. Specifically, **Emre** was tasked to develop the code for the detection of outliers and the creation of multivariate dataset, while **Hui** was assigned to develop the code to generate a multivariate dataset and **Eamonn**'s task was to develop the code to collect and visualise the multivariate dataset training. In terms of the report, Emre contributed to the Outliers section while Hui and Eamonn contributed to the Multi-collinearity section of the report.

Following this work allocation, each member worked individually and separately from each other on the respective tasks. However, we soon realised that there was considerable duplication of work, and therefore an unnecessary waste of time and resources.

We then decided that it was more productive and fruitful to work together, where ideas could be shared and issues worked out, with substantial inputs from each team member towards the tasks. This was achieved through weekly meetings, either online through Teams, or physically at the University premises.

This approach allowed everyone to collaborate with each other, and provide help, suggestions, inputs to all the tasks. Working this way means that everyone contributed to every part of the assignment. This is also in line with the coursework guidelines that we were expected to work together on the task problem instead of dividing the work out.

As such, the final output merges all the contributions of the individual team members and presents the product of our team working synergistically together.