|  |
| --- |
|  |
| Programming & Maths for AI – Task 2 |
|  |

|  |
| --- |
|  |

Table of Contents

[Introduction 2](#_Toc90087675)

[Outliers 2](#_Toc90087676)

# Introduction

For task 2, we will be analysing how the **presence of outliers** and the **presence of selection bias** affects the performance of a Linear Regression model. We will be looking at different performance factors such as the models:

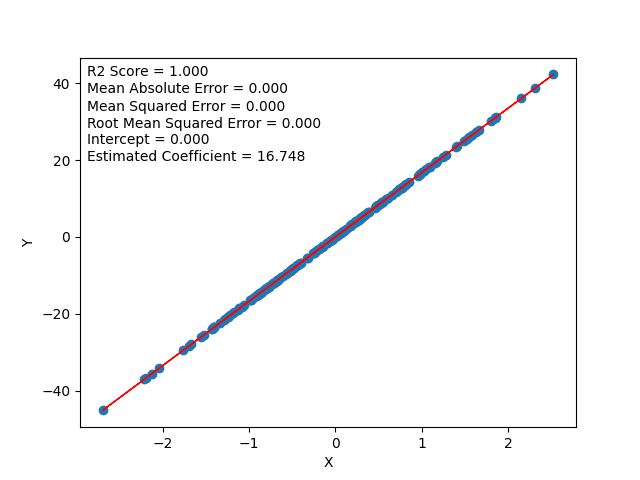
* **Coefficient of Determination**
* **Mean Absolute Error**
* **Mean Squared Error**
* **Root Mean Squared Error**

# Outliers

For outliers, we will be examining how to **number of outliers** and the **degree** (standard deviation) **of outliers**.

We will go about testing this like the following: 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.

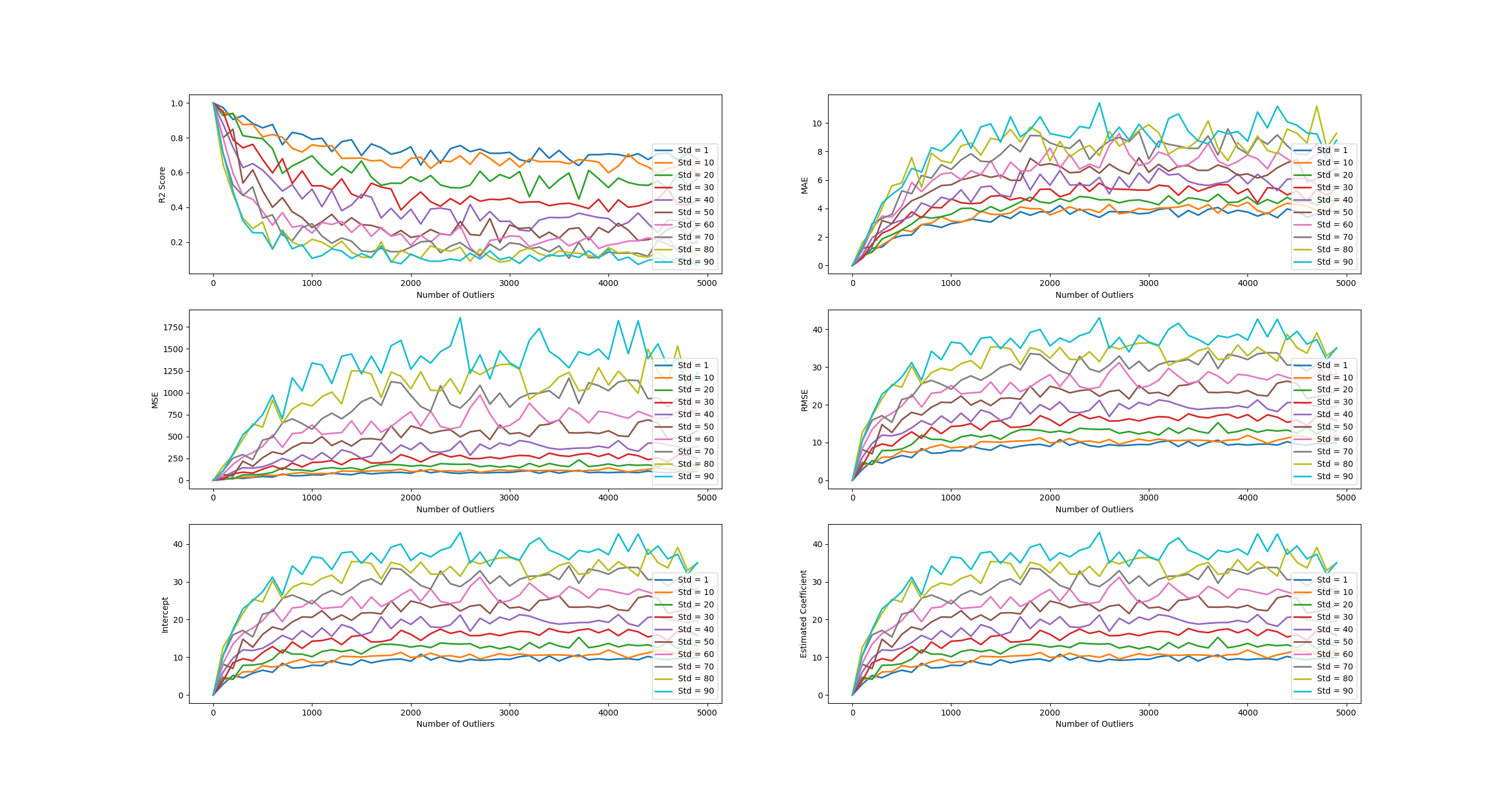
The model in all cases will be trained using Sklearn’s **LinearRegression** Model with the default parameters. The data used to train the model will be split using a standard 80-20 train-test split. The number of samples used for the clean dataset is 5000. An example of what outliers added to this clean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

Chart, scatter chart

Description automatically generated

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.

Chart, diagram

Description automatically generated

A 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 understanding the theory, the results align with what we would expect 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 you decided you cannot remove the outlier, but it is having a negative impact on the model, you 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**. You could also **impute** the outliers if you decided 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 you have 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, you could try some of the following options:

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.