A sensitivity analysis of different machine learning methods

Platon Frolov
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
p.m.frolov@student.utwente.nl

ABSTRACT

Datasets often contain noisy data due to faulty calibration of sensors or human error. A substantial amount of research has been conducted on the impact of noise on the accuracy of models to provide explainability of these so-called black-box models. However, less research has been conducted on how the noise impacts the precision of the models, which could provide an additional dimension of explainability about the robustness of the models. This paper provides insight into the robustness and explainability of machine learning regression methods by looking at what the influence of perturbations in numerical features in training data is on the variance of the output of linear regression, regression trees and multi-layer perceptron regression methods. The research has been conducted with an experimental approach in which the regression methods were exposed to different variances in Gaussian noise added to attributes in the training dataset. From the experiments, it appeared that decision trees are notably more sensitive to attribute noise than linear regression, and multi-layer perceptron regression. The latter two methods show a high tolerance to noise in the training data on the specific datasets.

Keywords

Sensitivity, Multi-layer perceptrons regression, Tree regression, Linear regression, Gaussian noise, Precision

1. INTRODUCTION

Over the past years, the tasks that machine learning methods have to perform have become increasingly large and complex. When machine learning methods such as neural networks and decision trees become large in size, they are very hard for humans to interpret as they are highly recursive and too big to visualize properly [15]. Attempts to make the models more interpretable for humans have been made by researching feature importance in linear regression, regression trees and multi-layer perceptron regression [2, 6, 7]; which feature x_i has the most influence on the output y and how much does each feature contribute to the prediction of the output y? This type of research provides insight into how a model reasons and which features could be discarded due to irrelevance. However, it does not provide enough explainability, for example, about the robustness or sensitivity to noise of the methods; how much does the variance in a feature contribute to the variance in the output?

Since machine learning methods are used more and more nowadays, also in critical fields such as healthcare and criminal justice, more transparency is required. The lack of transparency and robustness of predictive models can deeply impact human lives [15]. As noise is very common in real-world data, it is important to evaluate what the impact of noise is on the machine learning methods and how robust they are to noise because the noise could lead to inaccuracy, inconsistency and wrong predictions.

Contribution

The main contribution of this paper is to investigate how the preciseness of linear regression, regression trees and neural networks is influenced by Gaussian noise in the training data. In particular, the methods will be tested against Gaussian noise with different variance in continuous numerical attributes. The following research question will be answered:

In order to get more insight into the robustness and explainability of machine learning regression methods, what is the influence of different magnitudes of variances in perturbations in numerical features in training data on the precision of linear regression, tree regression, and multi-layer perceptron regression methods?

To answer this question, an experimental approach was taken[1]. The selected datasets were corrupted by adding Gaussian noise to the datasets in a controlled manner. For different variances, the variance in the output of the models, or in other words, the precision was evaluated. A more detailed description of the methodology can be found in Section 4. Section 5 and 6 contain the results and discussion respectively. But first, background theory and related work are given in Section 2 and 3.

2. BACKGROUND

2.1 Machine learning regression methods

2.1.1 Linear regression

Linear regression is used to solve regression problems by a predictive linear model. In multiple linear regression, the goal is to make predictions of a regression variable, the predictor variable, from one or more quantitative attributes,

[1]The code used for the experiments can be found at: `https://github.com/platonfrolov/research_project`

35th Twente Student Conference on IT July 2nd, 2021, Enschede, The Netherlands.

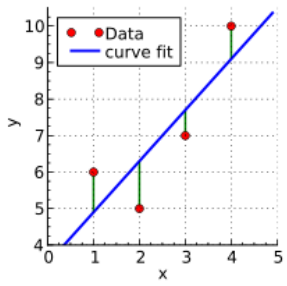


Figure 1: Visualization of the least squares method [25].

the explanatory variables [22, p.6]. Linear regression assumes a linear relation between the explanatory variables and the predictor variable [13]. If the k explanatory variables would be called $X = \{X_0, X_1, \cdots, X_k\}$ and the predictor variable Y, then the model can be described as in Equation 1.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} \tag{1}$$

Where each $\beta_j, j \in \{1, \cdots, k\}$ is determined by the method of least squares. This means that the β's are chosen in such a way that the line approximating the relation has the minimum sum of squares of the vertical distances between each observation point and the line. In Figure 1, a visualization of how the least squares method works is shown. The line is drawn in such a way that the sum of the squares of the length of the green lines is minimal. During training, the model is evaluated with the Residual Sum of Squares method (RSS). The formula to calculate the residual sum of squares is as follows:

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Linear regression is computationally cheap but it has one major drawback; if the datasets are not modelled adequately by a linear function, then linear regression is not very accurate as it assumes linearity.

2.1.2 Regression trees

Regression trees are a subset of decision trees, which is a supervised learning technique used to solve regression problems. In regression trees, predictions are made based upon learning decisions that are derived from features in the training dataset. When the tree is built from the data, an unseen data point can be put into the model to predict the predictor variable by traversing the tree until one ends up in a leaf node. The leaf node contains the numerical value prediction. In Figure 2, a visualization of how a regression tree comes to its predictions based on splitting criteria is shown.

When building a decision tree, multiple methods for determining the next split can be used. For classification trees, criteria such as information gain, the Gini index and Chi-square are used. This research is restricted to the use of the Mean Squared Error (MSE) criterion to build the regression trees. The mean squared error of a split can be calculated through the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - r(\beta, \mathbf{x}_i))^2$$

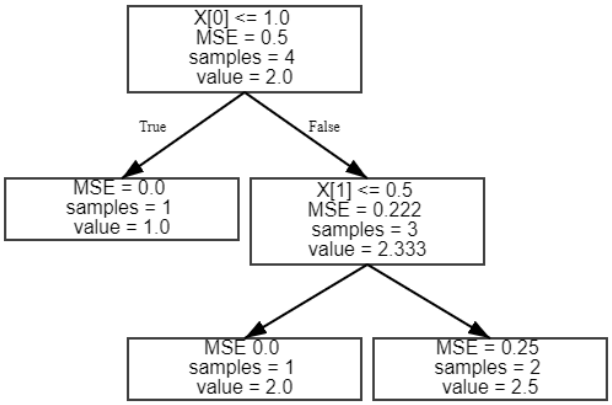


Figure 2: A visualization of a decision tree and its decision nodes.

Where $r(\beta, \mathbf{x}_i)$ is the prediction of the regression model $r(\beta, \mathbf{x})$ for the case $(\mathbf{x}_i, y_i)$. In the process of building a tree with the mean squared error criterion, for each variable, for each possible value of that variable, the data is split into subsets. Subsequently, the MSE of each split is calculated. The variable together with the value that produces the most different subsets, or in other words, produces the smallest MSE, becomes the new splitting criterion. This is done recursively on each subset until the specified maximum depth is reached or until no further split is possible.

Regression trees generally do not work very well with continuous numerical values as a small change in a value leads to a big change in the tree, causing instability. Out of the three methods, they are easily interpretable by humans and they are computationally relatively cheap. On smaller datasets with fewer rows and attributes, they are prone to overfitting and there are limitations on the functions they try to approximate [1, 21].

2.1.3 Multi-layer perceptron regression

Multi-Layer Perceptrons (MLP) are a subset of neural networks. Each node in the perceptron has inputs with weights and an activation function to produce an output. The activation function is a transformation that transforms the output of a node before it is sent to the next layer of nodes. Each output y of a node, including the transformation of the activation function, can be calculated with the following function:

$$y = \phi(\sum_{i=1}^{n} w_i x_i + b)$$

Where ϕ is the activation function, x_i the feature vector, w_i the weights, and b the bias. The activation function that will be used in this research is the Rectified Linear Unit (ReLU) activation function, which is the standard for MLP regressors in the framework we use. The ReLU function is shown in Equation 2.

$$\phi(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{2}$$

MLPs are a subset of deep artificial neural networks, which means that they have multiple layers between the input

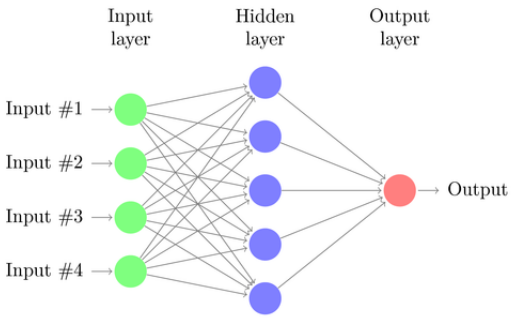


Figure 3: A visualization of an MLP with one hidden layer [16].

and output layer. Furthermore, MLPs are feedforward networks, implying their graph representations are acyclic. In Figure 3, a visualization of an MLP with one intermediate layer, also called a hidden layer, can be found. As can be seen in Figure 3, an MLP consists of an input layer, an output layer, and at least one intermediate hidden layer. These layers are the computational kernel of the MLP [20]. During the training of the model based on the training data, each row of the training data is fed into the model one by one. After each entry, the output of the network is compared with the actual value. Afterwards, the error is propagated back into the network to change the weight in such a way that the model fits better on the training data.

MLPs are the most complex models out of the three discussed in this paper and their predictive capabilities are more sophisticated than linear regression methods as they can identify non-linear relations between the variables. As a result, they are computationally heavier than the other methods discussed in this paper. Attempts have been made to remove redundant edges and nodes to reduce network complexity, but in this paper, no optimizations are used.

2.2 Gaussian noise

In this research, datasets will be corrupted with Gaussian noise. Gaussian noise has been chosen because it can be used to model processes that are subject to the Central limit theorem. The central limit theorem can be used almost everywhere [3]. Gaussian noise is statistical noise with the probability density function identical to the normal distribution. The probability of the noise having a value of y with a mean μ, and standard deviation σ is given in Equation 3 [9]:

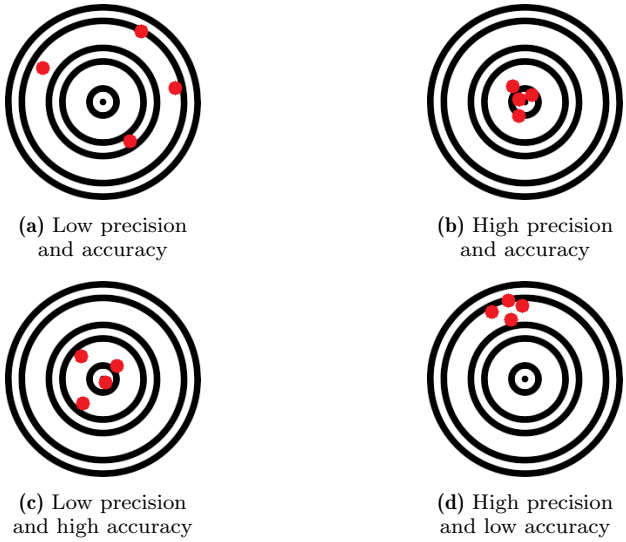
$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y-\mu)2}{2\sigma^2}} \tag{3}$$

The noise will be added to certain numerical features in the training data. Let $X = \{X_1, X_2, \cdots, X_n\}$, be a dataset and $x = \{x_1, \cdots, x_n\}$ an entry in the dataset. If we want to corrupt features k and l, $k, l \in \{1, \cdots, n\}$ in x with Gaussian noise, then the corrupted entry $\tilde{x}$ looks as in Equation 4.

$$\tilde{x} = \{x_1, \cdots, x_k + \epsilon_1, \cdots, x_l + \epsilon_2, \cdots, x_n\} \tag{4}$$

Where $\epsilon_1, \epsilon_2 \sim N(0, \sigma^2)$

2.3 Precision and accuracy



(a) Low precision and accuracy

(b) High precision and accuracy

(c) Low precision and high accuracy

(d) High precision and low accuracy

Figure 4: A visualization of the difference between precision and accuracy (with respect to the middle).

Precision refers to the closeness of two or more measurements. In this research, the precision of the three machine learning methods will be evaluated when noise is present in the datasets. Accuracy, however, is not taken into account and we leave this exploration to future research. Both accuracy and precision reflect how close two or more values are to each other relatively. Accuracy evaluates how close a measurement is to a known or true value, whereas precision evaluates the closeness of two measurements. This means that in order to be precise, one does not need to be accurate. The difference between precision and accuracy is illustrated in Figure 4.

The variance is a statistical measure that measures the dispersion in a set of numbers [12, p.29]. In other words, it gives an indication of how close the numbers are to each other. For this reason, it is the reciprocal of precision. Therefore, the variance of the outputs is used as a metric for precision in this research. The variance (σ^2), given measurements $\{x_1, x_2, \cdots, x_n\}$ can be calculated by:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

Where $\bar{x}$ is the mean value of the observations.

3. RELATED WORK

Previous work has shown that attribute noise could have severe consequences for the predictions of models [26]. Therefore research has been conducted that proposed new methods on how to clean datasets from noise [11, 24, 18].

Furthermore, research has been done on the sensitivity of the models to noise. In 2020, Schooltink performed a sensitivity analysis of support vector machine and random forest classifiers [17]. The metric for measuring the sensitivity was the accuracy of the classifiers. The experiments were performed with different levels of noise, ranging from 0% to 100% in the training data of the models. It appeared that both machine learning methods have a high tolerance for noise in the datasets that were used for the experiment up until a certain point. After a certain level

of noise, the accuracy of the methods decreased rapidly. In 2021, similar research was conducted by Stribos, who performed research on the impact of different types of noise on Naive Bayes classifiers [19]. First, the effect of test data noise and training data noise was evaluated. Second, the impact of class noise and attribute noise were compared against each other. Last, random noise was compared against structural noise.

In 2010, Nettleton performed research on the influence of different types of noise on the accuracy of different machine learning classifiers such as naive Bayes, decision trees with pruning (C4.5) and support vector machines [10]. The research showed that naive Bayes and C4.5 were quite robust methods to noise, whereas support vector machines showed some weaknesses.

In contrast to the aforementioned research, this paper aims to investigate the sensitivity to Gaussian noise of different regression methods such as MLPs, regression trees and linear regression. Furthermore, not accuracy, but precision of the methods will be taken as the metric for sensitivity.

4. METHODOLOGY

4.1 Selecting datasets

For this research project, we selected three datasets from the UCI machine learning repository[2] to test the sensitivity, or more specifically, the preciseness, of the three methods. The datasets were chosen because the prediction attributes of the datasets are numerical, making them regression problems. Furthermore, all datasets contain at least three numerical features in the training data, which will be corrupted by adding Gaussian noise before training a model on the data.

The first dataset contains data about the specifications and performances of different types of cars [23]. From this data, the fuel consumption of the cars in miles per gallon can be predicted, which is a continuous, numerical attribute. The second dataset consists of data about the time, the date, and the weather conditions and descriptions around a metro station in the US [8]. With this data, the hourly westbound traffic volume can be predicted, which is a continuous numerical feature. The third dataset holds all kinds of demographic data, such as the race composition and data about poverty and wealth in different populations [14]. With this data, the number of violent crimes per capita can be predicted, which is also a continuous numerical variable. A brief overview of the datasets can be found in Table 2.

4.2 Pre-processing the datasets

After selection of the datasets, the datasets were not ready for a model to be fitted on the training data because there were many missing values, too many rows and different scales for every feature. Therefore, we discarded rows with random missing values and columns with more than 50% of missing values. Next, we encoded categorical variables into integers, representing the category. And lastly, since the attributes in the datasets all had a different scale and unit, all values were standardized to have a value between 0 and 1 (min-max-scaling [4]) to get rid of the different scales. For the metro dataset, a random subset of 500 instances was taken to reduce the training times in order to finish the experiment within a reasonable amount of time.

[2]https://archive.ics.uci.edu/ml/index.php

Dataset	Selected features
Car	horsepower, weight, acceleration
Metro	temperature, rain_1h, snow_1h
Community	PctPopUnderPov, PctPopBlack, medIncome

Table 1: The selected features per dataset.

The community data set contains 1993 entries, as can be seen in Table 2. We did not reduce the number of entries in order to test the methods on datasets of different sizes. From all the datasets, three numerical features were selected to which noise would be added in the training data. Furthermore, the data was split up into training data and test data, in an 80/20 ratio respectively. The test data remained untouched throughout the whole experiment.

4.3 Corrupting the datasets

To answer the research question, we increased the variance of the noise added and observed the effects. The variance of the noise differed from 0 to 0.15 with steps of 0.01. All other variables were fixed. Each feature combination of the three selected features was corrupted twenty times with random Gaussian noise from the same distribution, resulting in 20 slightly differing training datasets. Only 10% of the entries were corrupted with noise. The 10% was not arbitrarily chosen. It appeared through trial and error that the output variance was not extreme but also not insignificant with 10% of the features corrupted. In Table 1, the three selected features are listed. These specific features were chosen because they are continuous numerical attributes.

4.4 Experimental setup

In order to evaluate the precision of the methods, we trained the models multiple times on training data with random Gaussian noise, with the same variance, in a subset of attributes and evaluated the dispersion in the output of the models trained on noisy data. We repeated this experiment for different variances in the Gaussian noise. More specifically, the noise was distributed by a Gaussian distribution with mean 0 and variance ranging from 0 to 0.15 with steps of 0.01. For each variance, the training data was corrupted 20 times with the noise added to 10% of the entries in the training data. Subsequently, 20 models were trained on these corrupted training datasets, with a random seed so that the results are reproducible. The (untouched) test data was used to make predictions with all 20 models. The mean output variance of the predictions was calculated and a point in the graph could be plotted where the variance of the noise represented the x-axis and the mean output variance represented the y-axis. The pseudo-code for the algorithm to generate a single point in the plot is shown in Algorithm 1. By performing Algorithm 1 with different variances, a line can be drawn through all the points.

Algorithm 1: Partial experiment algorithm

```
Result: Point in a plot
training data;
test data;
for 20 times do
    corrupt training data with noise from N(0, σ²);
    train model with corrupted data;
    evaluate model on test data;
end
output variance = variance of 20 evaluations;
plot(variance, output variance);
```

	Car dataset	Metro dataset	Community dataset
Number of attributes (Columns)	8	9	128
Number of instances (Rows)	398	48204	1994
Missing values	Yes	No	Yes
Columns >50% of entries missing	No	No	Yes
Normalized data	No	No	Yes
Categorical variables	Yes	Yes	Yes
Rows after preprocessing	392	500	1993
Columns after preprocessing	8	12	104

Table 2: Brief overview of all datasets and their properties.

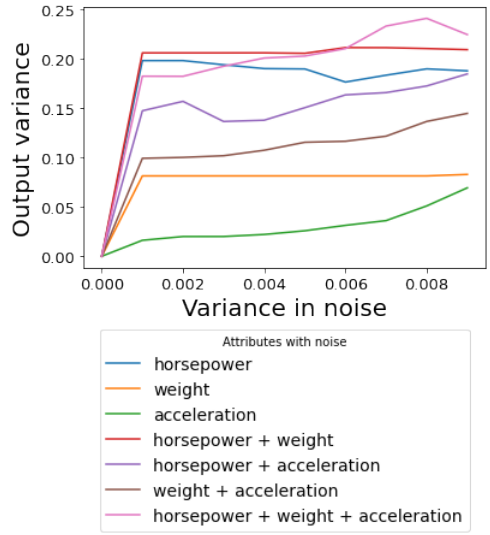


Figure 5: Results of the additional experiment for regression trees with noise variance ranging from 0 to 0.01 with steps of 0.001, added to 10% of the entries of the attributes with noise in the car training data set.

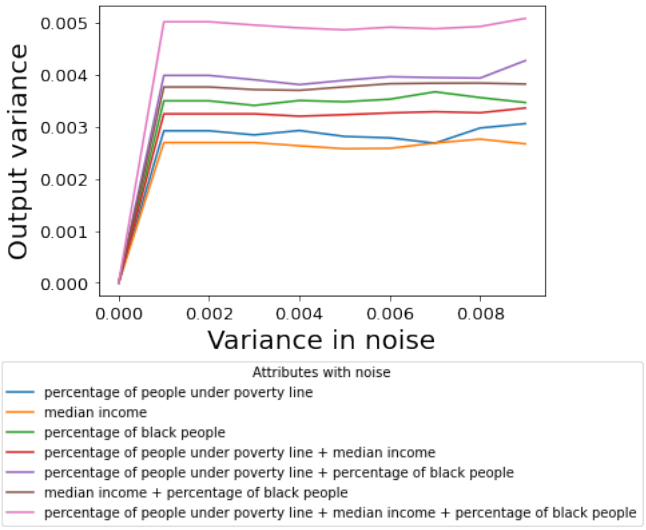


Figure 6: Results of the additional experiment for regression trees with noise variance ranging from 0 to 0.01 with steps of 0.001, added to 10% of the entries of the attributes with noise in the car training data set.

Furthermore, noise was added to multiple feature subsets, which resulted in multiple lines in one plot. The noise was only added to a small subset of numerical features. For each combination of the three selected attributes, excluding the empty set, the Gaussian noise was added to 10% of the entries of the feature(s). This means that 7, $2^3 - 1 = 7$, lines can be drawn in the plot where each line represents how the mean output variance is related to the variance of the noise added to the specific feature (subset). For each machine learning method, the above experiment was performed on all three datasets.

During the evaluation of the plots, we stumbled upon a few interesting findings. In order to look deeper into those phenomena, we ran a few extra experiments. The experiments are conducted in the same fashion, but with a different range and step size of variances in the noise. This is done to be able to zoom in on an interval or look at a broader interval of noise variance.

5. RESULTS

The results of the experiments can be found in Figures 7, 8 and 9. Each figure represents a dataset and contains three plots, each representing the performance of the machine learning method on the dataset. Subfigures a, b, c represent linear regression, regression trees, and MLP regression respectively. The mean output variance is plotted against the variance in the Gaussian noise that is added to 10% of the entries. This allows for evaluating the precision of the models because the variance is the reciprocal of precision by definition.

For all machine learning methods, on all datasets, a common and expected pattern in the results can be seen; The larger the variance in the noise becomes, the larger the mean output variance becomes. Moreover, noise in certain features led to more variance than noise in other features. Generally, it can be seen that the larger the number of features corrupted, the more imprecise the models become.

Overall, the linear regression method acts the most predictable because a simple smooth line can be drawn through all the results and the results can be interpolated and extrapolated easily on all intervals. The results of the decision trees and neural networks are less predictable as the lines often contain spikes and a larger variance could yield a lower output variance as can been seen in figures 7b,c, 8b,c and 9b,c.

Furthermore, linear regression has the smallest mean output variance, hence, the best precision, in all cases followed by neural networks. The precision of decision trees is in all cases worse than the precision of the other methods.

5.1 Regression trees

For regression trees, an interesting phenomenon on all three datasets can be observed. As soon as any noise

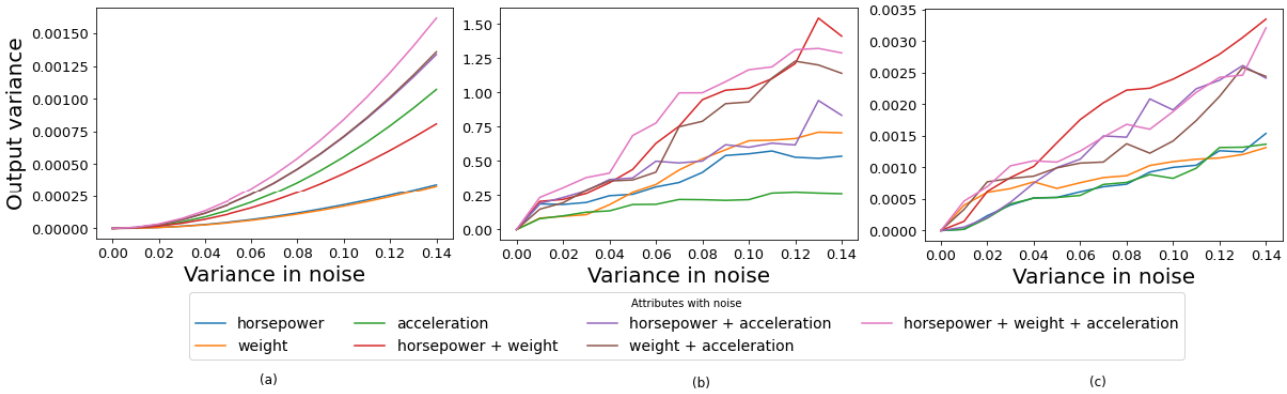


Figure 7: Results of the experiment with noise variance ranging from 0 to 0.15 with steps of 0.01, added to 10% of the training data in the car dataset. (a) linear regression, (b) tree regression, (c) MLP regression.

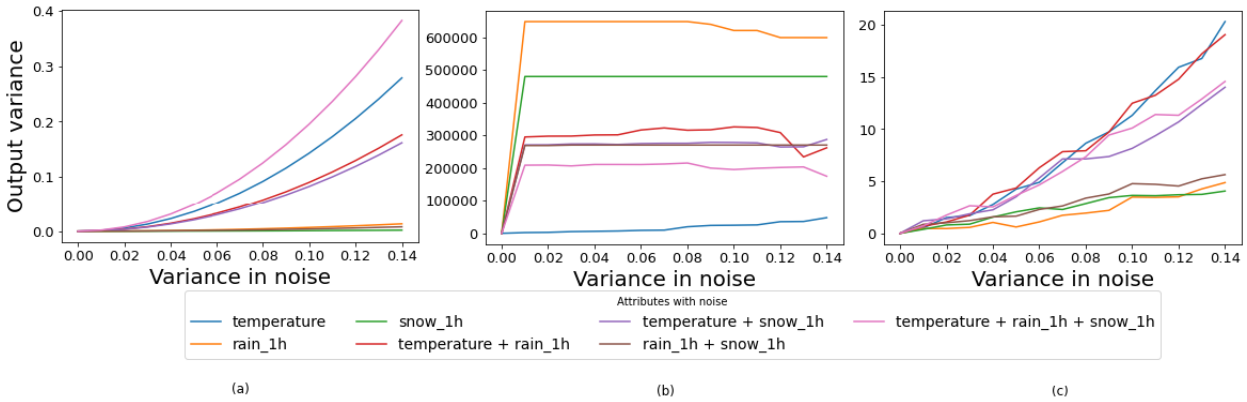


Figure 8: Results of the experiment with noise variance ranging from 0 to 0.15 with steps of 0.01, added to 10% of the training data in the metro dataset. (a) linear regression, (b) tree regression, (c) MLP regression.

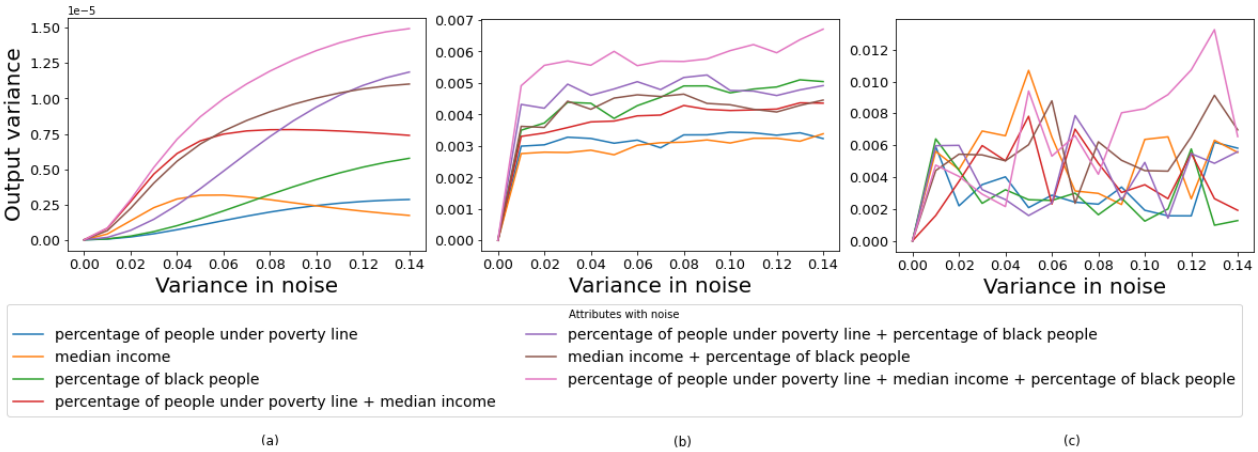


Figure 9: Results of the experiment with noise variance ranging from 0 to 0.15 with steps of 0.01, added to 10% of the training data in the community dataset. (a) linear regression, (b) tree regression, (c) MLP regression.

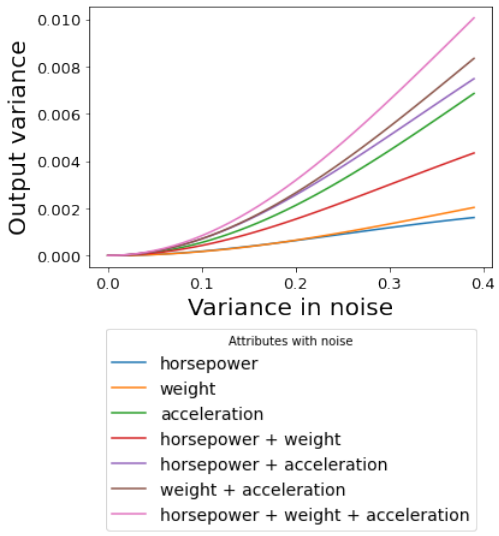


Figure 10: Results of the additional experiment for linear regression with Gaussian noise variance ranging from 0 to 0.4 with steps of 0.01, added to 10% of the entries of the attributes with noise in the car training data set.

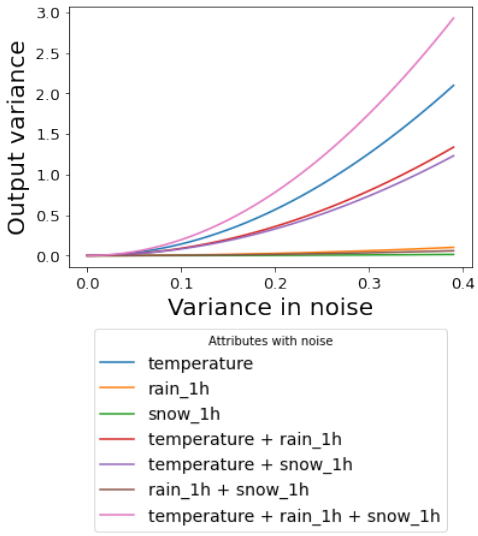


Figure 11: Results of the additional experiment for linear regression with Gaussian noise variance ranging from 0 to 0.4 with steps of 0.01, added to 10% of the entries of the attributes with noise in the metro training data set.

gets added to any combination of features, the mean output variance makes a large jump. Increasing the variance further does not have much impact on the precision of the model relative to smaller variances. In the interval $[0, 0.01]$, a huge drop in precision for regression trees can be seen on all data sets. Therefore, a more detailed look is taken at this interval. In Figures 5 and 6, the results of further investigation into this phenomenon have been plotted. Figure 5 shows how the output variance of the decision tree model behaves when the variance of the noise in the car dataset is in the interval $[0, 0.01]$ (with steps of 0.001). Figure 6 does the same for the community dataset. From these figures, it appears that regression trees are sensitive to very small corruptions in the datasets and that even from noise with a variance of 0.001, the model drastically loses precision. The zoomed-in plots (Figure 5 and 6) also suggest that the precision of the regression tree model deteriorates very quickly but adding any larger noise to the training data does not make the precision worse. This can be explained by the fact that still 10% of the entries is corrupted with small noise. In [5], the experiments were corrupted with increasing percentage (from 0% to 10% with steps of 1%) of corrupted entries and a fixed variance of 0.001 in order to see how it got to the jump. A more gradual increase in output variance can be observed, which means that the jump from 0 records to 10% of the records corrupted caused the large increase in output variance.

5.2 Linear regression

For linear regression, the relation between the noise variance and the output variance on the car and metro dataset seems to have a quadratic or exponential relation. To investigate this further, an additional experiment on the car and metro dataset has been performed. The extra experiment involves looking at how the output variance develops when the noise variance is larger than 0.15. Figures 10 and 11 show the output variance against a noise variance ranging from 0 to 0.4, which is an extension of the original interval. In the car dataset in Figure 10, it appears that from a noise variance of 0.2 or larger, the output variance has a linear relationship with the noise variance, and a quadratic/exponential relationship if the noise variance is smaller than 0.2. However, on the metro dataset (Figure 11) it appears that the relation between the noise variance and the output variance is quadratic/exponential over the whole interval [0, 0.4].

For the community dataset, it seems that at some point a maximum output variance is reached for the linear regression method unlike for the other datasets, as can be seen in Figure 9a. For linear regression most feature subsets are not very sensitive to increasing variances after a noise variance of 0.04-0.05 on this specific dataset. The output variance stabilizes and does neither decrease nor increase when making the variance in the noise larger.

5.3 Multi-layer perceptrons

The relations between the mean output variance and the variance in the noise seem to have an approximately linear relation when MLPs are used since a straight line could be drawn to approximate the lines in the neural network plots. However, as can be seen in figures 7c, 8c, 9c, the lines are non-smooth and some spikes regularly occur. More prominent noise in the training data sometimes leads to higher precision.

6. DISCUSSION

From the results it appeared that the larger the variance in the noise became, the larger the output variance became in general. This is expected because the larger variance in the noise ensures that the training datasets become more different than the other nineteen training datasets which are also corrupted with noise from the same distribution. This results in twenty more heavily contrasting models which will produce more varying outputs, resulting in a larger variance, hence, a worse precision.

In this research, the broad term linear regression, regression trees and MLPs are used, but as discussed in Section 2, we used very specific instances of these classes. Other instances with different activation functions, split-

ting functions, pruning, and parameters could lead to different results and insights.

6.1 Regression trees

Another thing that stood out from the results is that on all three datasets, the regression trees performed the worst. This can be explained by the fact that regression trees are very unstable as discussed in Section 2.1.2, especially when there are a small number of attributes and training instances and the trees are not pruned. A small change in the training dataset might lead to a different choice in a node and therefore the whole decision tree might end up looking completely different. Furthermore, the functions that regression trees approximate are very non-smooth and discrete. Which could lead to very prominent function discontinuities and jumps and therefore also large differences in the output.

In the community dataset, the performance of decision trees, relative to the other methods, is better than on the other datasets. As mentioned in Section 2.1.2, decision trees perform are sensitive and perform worse precision- and accuracy-wise when the number of entries is low [1]. Since the community dataset contains significantly more attributes and has the most rows, it is expected that decision trees are less sensitive to noise in the training data and therefore more precise on this dataset.

6.2 Linear regression

Linear regression performs the best when looking at the precision, followed by MLPs, which don't perform much worse. When fitting a model on a training dataset with some noise, the method of the least squares will yield slightly different coefficients β_i of the i^{th} attribute. Since the coefficients only differ slightly and the relation is linear, the differences will also be very slight. This makes it seem that linear regression models are very good, but there are restrictions on the use of linear regression. The accuracy of the linear regression models might be very poor if the relationship between the explanatory variables and the predictor variable is not approximable by a linear function.

6.3 Multi-layer perceptrons

The plots of the MLPs have a lot of spikes in them. This could be explained by the fact that MLPs are very complex models as stated in Section 2.1.3. It could be that a certain value or range of values of a feature leads to a large change in the output, whereas a value of an attribute in another interval does not change the output much. When adding greater noise to the training data, the value might be in a range where the output is not much different than the original, whereas the value with smaller noise changes the output of the model drastically as it is in the sensitive interval. Hence, the spikes in graphs and the uncertainty in precision.

7. CONCLUSION

The research performed in this paper showed that the precision of all methods decreases as the noise in the features becomes larger. Furthermore, the more features contained noise, the more the precision decreases. Noise in certain features caused a more drastic loss of precision than noise in other features. These features have the most influence on the output and can be considered more important to the prediction. Overall, linear regression has the best precision with attribute noise, followed by MLP regression. Tree regression performs significantly worse precision-wise than the other methods.

The variance of the outputs of the linear regression models is approximately linearly correlated with the variance in the training data noise. The method has proven to be very stable and predictable when noise is present in numerical features in the training data. The method is robust and suited for situations in which stability and certainty are required. However, the use of linear regression should be carefully considered as the method will have poor accuracy on data sets that cannot be modelled linearly.

Regression trees are very unstable when noise is added to the datasets, especially if the training dataset does not contain many entries and attributes. Even with slight noise, the precision decreases drastically. However, when increasing the variance in the noise, the precision remains approximately constant. Regression trees could be very useful in situations where the noise in the training dataset is unknown, and a lower precision is permissible. The method is not very robust to noise and does not provide a lot of explainability about the behaviour of the models.

The performance of MLPs was slightly worse than for linear regression. The precision was lower and the method acts more unpredictable, greater noise could sometimes yield higher precision. Overall, the method is quite stable and it could be used in situations where preciseness, also with noisy data is required as the method is quite robust. A major benefit of neural networks is that they are not restricted to having a high accuracy only for linear relations, but also non-linear ones.

8. FUTURE WORK

This research was restricted to only test a few machine learning methods on very specific datasets and noise. It is desirable to extend this research by researching other machine learning regression methods. Furthermore, instead of comparing different methods against each other, it might also be interesting to compare the methods against themselves with different splitting functions, activation functions, and other parameters. For a sensitivity analysis, instead of focusing on the precision of the methods, one could also look into the accuracy of the methods and eventually combine the two metrics to make conclusions about the sensitivity to noise in the training data.

Decision trees and neural networks are also widely used for classification purposes. Instead of testing the machine learning methods in a regression problem environment, one could also investigate the methods on classification problems. Moreover, this research aimed to test the methods with Gaussian noise in the continuous numerical attributes of the training data. However, different types of noise in different types of attributes could also be looked at in future work.

9. ACKNOWLEDGEMENTS

I would like to thank my supervisor Maurice van Keulen for guiding me through this project by giving feedback and directions. Furthermore, I would like to thank a few of my fellow students and my girlfriend for proofreading the paper.

10. REFERENCES

[1] J. Ali, R. Khan, N. Ahmad, and I. Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[2] A. Altmann, L. Tolo¸si, T. Tolo¸si, O. Sander, and T. Lengauer. Data and text mining permutation importance: a corrected feature importance measure. 26:1340–1347, 2010.

[3] G. A. Brosamler. An almost everywhere central limit theorem. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 104, pages 561–574. Cambridge University Press, 1988.

[4] X. H. Cao, I. Stojkovic, and Z. Obradovic. A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC bioinformatics*, 17(1):1–10, 2016.

[5] P. Frolov. Additional experiment results. `https://figshare.com/articles/figure/Additional_experiment_results/14853879/2`, Jun 2021.

[6] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. volume 33, pages 3681–3688. AAAI Press, 7 2019.

[7] J. Heaton, S. McElwee, J. Fraley, and J. Cannady. Early stabilizing feature importance for tensorflow deep neural networks. volume 2017-May, pages 4618–4624. Institute of Electrical and Electronics Engineers Inc., 6 2017.

[8] J. Hogue. Metro interstate traffic volume data set. `https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume`, 2019.

[9] Massachusets Institute of Technology. 6.02 lecture 9: Transmitting on a physical channel, 2011.

[10] D. F. Nettleton, A. Orriols-Puig, A. Fornells, D. F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif Intell Rev*, 33:275–306, 2010.

[11] F. Nijweide. Autoencoder-based cleaning of non-categorical data in probabilistic databases. *33rd Twente Student Conference on IT*, 2020.

[12] V. M. Panaretos. *Statistics for mathematicians: A rigorous first course*. 2016.

[13] M. A. Poole and P. N. O'farrell. The assumptions of the linear regression model, 1971.

[14] M. Redmond. Communities and crime data set. `https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`, 2009.

[15] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[16] Safdari Hartman. Neural networks for calibrating atlas jets, 2016. [Online; accessed June 09, 2021].

[17] W. Schooltink. Testing the sensitivity of machine learning classifiers to attribute noise in training data. *33rd Twente Student Conference on IT*, 2020.

[18] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[19] R. Stribos. The impact of data noise on a naive bayes classifier. *34th Twente Student Conference on IT*, 2021.

[20] H. Taud and J. Mas. *Multilayer Perceptron (MLP)*, pages 451–455. Springer International Publishing, Cham, 2018.

[21] L. F. R. A. Torgo. Inductive learning of tree-based regression models, 9 1999.

[22] M. Tranmer, J. Murphy, M. Elliot, and M. Pampaka. *Multiple Linear Regression (2 nd Edition)*. 2020.

[23] C. M. University. Auto mpg data set. `https://archive.ics.uci.edu/ml/datasets/Auto+MPG`, 1993.

[24] V. Vijaykumar, P. Vanathi, and P. Kanagasabapathy. Fast and efficient algorithm to remove gaussian noise in digital images. *IAENG International Journal of Computer Science*, 37(1):300–302, 2010.

[25] Wikipedia, the free encyclopedia. Linear least squares. `https://upload.wikimedia.org/wikipedia/commons/thumb/b/b0/Linear_least_squares_example2.svg/330px-Linear_least_squares_example2.svg.png`, 2021. [Online; accessed June 09, 2021].

[26] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210, 2004.