Impact of Noise in Dataset on Machine Learning Algorithms

Preprint · February 2019

DOI: 10.13140/RG.2.2.25669.91369

5 authors, including:

Arun Thundyill Saseendran
Trinity College Dublin
3 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Lovish Setia
Trinity College Dublin
2 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Viren Chhabria
Trinity College Dublin
2 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Debrup Chakraborty
Trinity College Dublin
2 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Impact of Noise in Dataset on Machine Learning Algorithms

Arun Thundyill Saseendran
School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin, Ireland
thundyia@tcd.ie

Lovish Setia
School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin, Ireland
setial@tcd.ie

Viren Chhabria
School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin, Ireland
chhabriv@tcd.ie

Debrup Chakraborty
School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin, Ireland
chakrabd@tcd.ie

Aneek Barman Roy
School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin, Ireland
barmanra@tcd.ie

ABSTRACT

Noise can have a significant impact on the overall performance of a machine learning model. This work studies the strength of impact of noise on regression (linear and polynomial regression with ridge regularization) and classification algorithms (random forest classifier). Based on the study, the authors observe that with reference to the performance of a model on a zero-noise preprocessed dataset for every ten percent increase in random sample noise, there is an increase of RMSE by 3.2 percent for polynomial and linear regression with ridge regularization machine learning algorithms. On the other hand, random forest classifier with reference to a model trained on a zero-noise preprocessed dataset suffers a decrease in accuracy of 3 percent with every ten percent increase in sample noise.

Keywords: Impact of Noise, Machine Learning, Noisy Data, Sample Noise, Feature Noise

INTRODUCTION

Meaningless or corrupted data in datasets is known as noise. This noise can result in errors in the predictions by the machine learning algorithms and can impact their performance in terms of accuracy, size of the model and the time taken to build the model [1].

This work analyses the strength of impact of noise on two types of machine learning algorithms; linear and polynomial regression algorithms and random forest classification algorithm with respect to performance in terms of accuracy.

1 RELATED WORK

Zhu and Wu have conducted a quantitative study of impact of noise on machine learning classification algorithms [1]. The authors have shown that with increase in feature noise the accuracy of the classification algorithms decreases linearly. In terms of attribute noise, the study shows that the lowest level of classification accuracy is given by classifiers trained using noisy dataset compared to clean dataset on both clean and noisy target and the performance deteriorates linearly with addition in noise.

In another study, the effect of noise on 4 classification algorithms with various degrees and types of noise is studied. Of the 4 algorithms used, the authors conclude that Naïve Bayes and C4.5 are resistant towards noise with the former being the strongest, and IBk and SMO the least resistant to noise with the latter the worst of all [2].

2 METHODOLOGY

2.1 Dataset Analysis and Preprocessing

Three datasets are used in this work: Bike renting [3], wine quality [4], and mobile price classification [5].

The scripts used in this work are implemented in Python 3.6 [6] using the Scikit-learn framework [7] and is maintained in github [8].

The workflow of data pre-processing and feature selection used in this work is shown in Figure 1.

The first step of the workflow is to visualize the data. Then the correlation matrix is created and features with very high or very low correlation are removed. After that the data is scaled using Robust Scaler. Robust scaler is chosen since it is based on percentiles and hence not influenced by a few numbers of large outliers [9]. The outlier detection is done based on Interquartile range (IQR). Handling of the outliers is done on a case to case basis and is retained, pruned or winsorized. Post outlier handling, the

importance of features is calculated using Random Forest Regressor and feature selection is done.

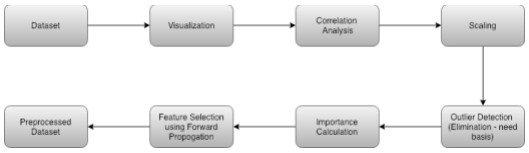


Figure 1: Data Preprocessing and Feature Selection Workflow

2.1.1 *Bike Dataset:* The bike dataset's [3] details are given in *Table 1* and features explained in *Table 2*.

Table 1: Details of Bike Dataset

Detail	Value
Attribute Characteristics	Integer, Real
Number of Instances	17389
Number of Attributes	16
Missing Values	0
Year of Data	2011-2012

Table 2: Feature Description of Bike Dataset

Feature Name	Description	Datatype
instant	Record index	Integer
dteday	Date	Date
season*	Season (1-Spring, 2-Summer, 3-Fall, 4-Autumn)	Integer
yr*	Year	Integer
mnth*	Month (1-12)	Integer
hr	Hour of the Day (0-23)	Integer
holiday	Whether the day is holiday or not (0,1)	Boolean
weekday	Day of the week (0-6)	Boolean
workingday	If the day is neither a holiday nor a weekend, 1 else 0	Boolean
weathersit	Weather Situation (1-Clear, 2-Mist, 3-Light Snow, 4-Heavy Rain)	Integer
temp*	Temperature in Celcius.	Float
atemp	Feeling Temperature in Celcius.	Float
hum*	Normalized humidity. The values are divided to 100 (max)	Float
windspeed*	Normalized wind speed. The values are divided to 67 (max)	Float
casual	Count of casual users	Integer
registered	Count of registered users	Integer
cnt	Target variable: Count of total rental bikes including both casual and registered	Integer

The correlation matrix for the bike dataset is show in Figure 2 and the importance of features in Figure 3. Features marked with '*' in *Table 2* were selected based on correlation and importance.

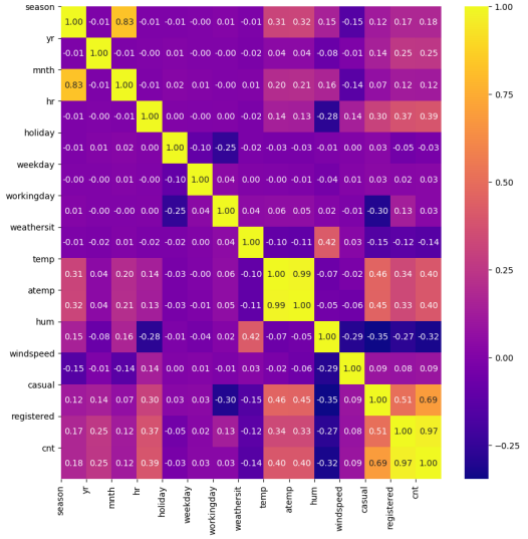


Figure 2: Correlation Matrix for Bike Dataset

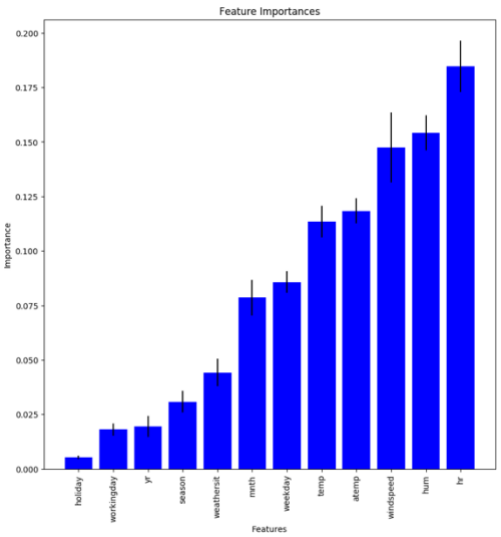


Figure 3: Feature Importance for the Bike Dataset

The histogram for the target variable 'cnt' and the IQR based box plot for outliers is show in Figure 4 and Figure 5 respectively. Though some features had outliers as per IQR, they were not pruned as the values were logically correct.

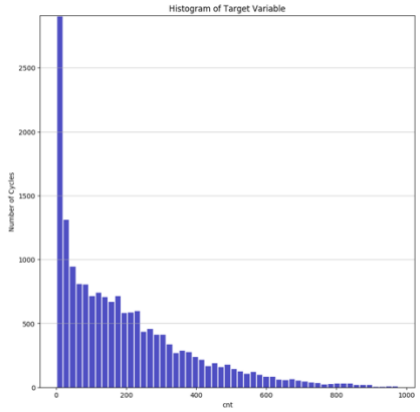


Figure 4: Histogram for target variable 'cnt' in Bike Dataset

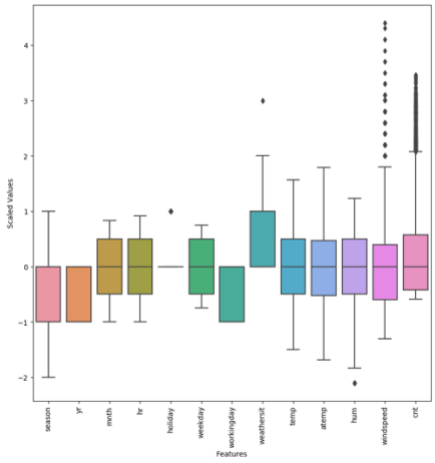


Figure 5: IQR based boxplot for features in Bike Dataset

2.1.2 Wine Dataset: The wine quality dataset's [4] details are given in *Table 3* and the features explained in *Table 4*.

Table 3: Details of Wine Dataset

Detail	Value
Attribute Characteristics	Integer
Number of Instances	3919
Number of Attributes	13
Missing Values	0
Year of Data	2009

Table 4: Feature Description of Wine Dataset

Feature Name	Description	Datatype
fixed.acidity*	Fixed acidity of wine	Integer
volatile.acidity*	Volatile acidity of wine	Integer
citric.acid	Citric acidity of wine	Integer
residual.sugar*	Residual sugar contained in wine	Integer
chlorides*	Chlorides contained in wine	Integer
free.sulfur.dioxide	Free Sulphur dioxide contained in wine	Integer
total.sulfur.dioxide*	Total Sulphur dioxide contained in the wine	Integer
density*	Density of wine w/v	Integer
pH*	ph measure of the wine	Integer
sulphates	Sulphates contained in the wine	Integer
alcohol*	Alcohol content in the wine v/v	Integer
quality*	Quality of the wine	Integer
id*	Unique identifier of the wine	Integer

The correlation matrix and the feature importance bar graph for the wine dataset is shown in Figure 6. Based on the correlation matrix and importance, the features marked with '*' in *Table 4* are selected.

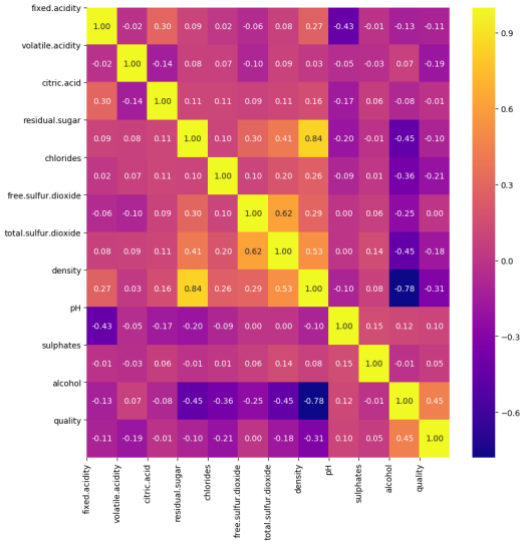


Figure 6: Correlation Matrix for Wine Dataset

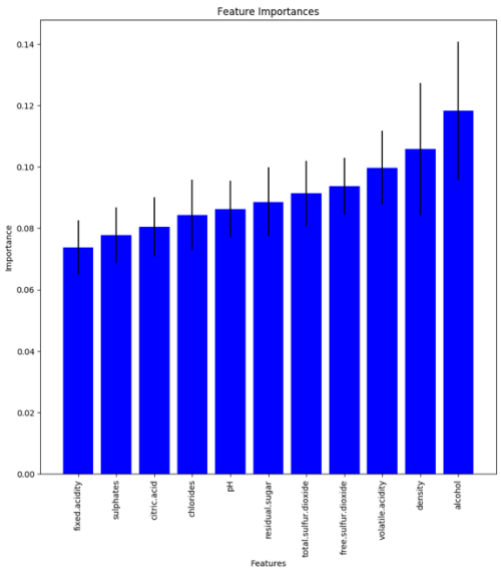


Figure 7: Feature Importance for the Wine Dataset

2.1.3 Mobile Pricing Dataset: The mobile pricing dataset [5] details are given in *Table 5* and the features explained in *Table 6*.

Table 5: Details of Mobile Pricing Dataset

Detail	Value
Attribute Characteristics	Integer
Number of Instances	2000
Number of Attributes	20
Missing Values	0
Year of Data	2017

Table 6: Feature Description of Mobile Pricing Dataset

Feature Name	Description	Datatype
id	ID	Integer
battery_power*	Total energy a battery can store in one time measured in mAh	Integer
blue	Has bluetooth or not	Boolean
clock_speed	speed at which microprocessor executes instructions	Float
dual_sim	Has dual sim support or not	Boolean
fc	Front Camera mega pixels	Boolean
four_g	Has 4G or not	Boolean
int_memory*	Internal Memory in Gigabytes	Integer
m_dep	Mobile Depth in cm	Float
mobile_wt*	Weight of mobile phone	Integer
n_cores	Number of cores of processor	Integer
pc*	Primary Camera mega pixels	Integer
px_height*	Pixel Resolution Height	Integer
px_width*	Pixel Resolution Width	Integer
ram*	Random Access Memory in Megabytes	Integer
sc_h	Screen Height of mobile in cm	Integer
sc_w	Screen Width of mobile in cm	Integer
talk_time	longest time that a single battery charge will last when you are	Integer
three_g	Has 3G or not	Boolean
touch_screen*	Has touch screen or not	Boolean
wifi	Has wifi or not	Boolean
price_range	This is the target variable with value of 0(low cost); 1(medium cost); 2(high cost) and 3(very high cost).	Category

The correlation matrix and the feature importance bar graph for the mobile pricing dataset is shown in Figure 8. Based on the correlation matrix and importance, the features marked with '*' in Table 6 are selected.

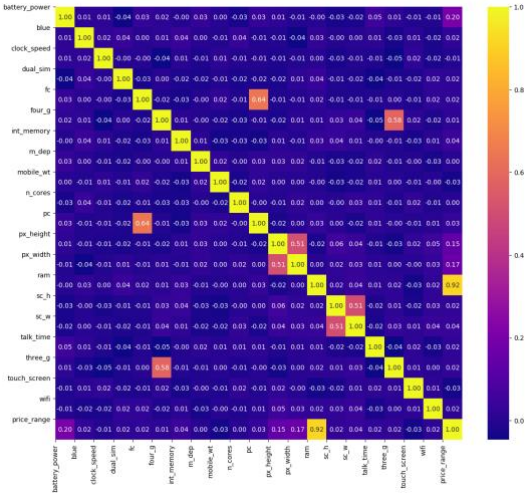


Figure 8: Correlation Matrix for Mobile Pricing Dataset

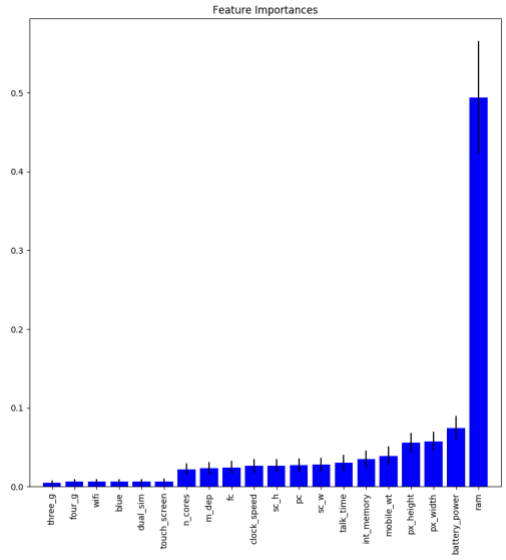


Figure 9: Feature Importance for the Mobile Pricing Dataset

Though there are outliers as per IQR, they are not handled since it was logically meaningful.

2.2 Noise Generation and Insertion

For experiments, multiple datasets are created from the original dataset by adding different levels and type of noise to them.

2.2.1 Sample Noise: For a dataset D of shape [x, y], weight scalar *k*, the percentage of noise n and number of samples *m*, the noisy dataset array *DA* with noise *N* of the shape as [z, y] is generated based on equation *(2)* . The addition of samples with respect to the original number of samples *x* is given in equation *(1)* .

$$z = x + (k * \mathrm{m}) \quad \textbf{(1)}$$

$$DA = set(\lim_{0 \to n} D + N[z, y]) \quad \textbf{(2)}$$

The sample noise is inserted per feature based on the datatype of the feature. For a feature *f* and a noise spread factor *s*, the noise data is created using a normal random distribution function with the lower and upper limits as defined in equation (3).

As the noise is evenly distributed among the samples, this will have smaller effect on datasets with larger number of samples rather than ones with lesser number of samples [10].

$$f_{rand}(\min(f) - (s/100), \max(f) + (s/100)) \quad \textbf{(3)}$$

An example with fifty percent noise added to one of the features (windspeed), in the bike dataset [3] is show in *Figure 10* .

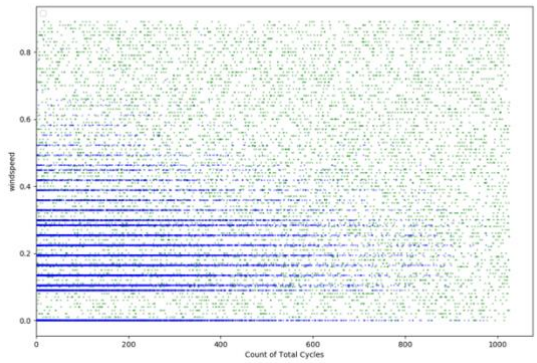


Figure 10: Example of fifty per cent noise added to the feature windspeed of the bike dataset where the blue dots are real data and green dots are noise

2.2.2 *Feature Noise:* The noise is added to the dataset by adding noisy features to the existing dataset. The noisy feature is generated by creating a new feature f_{noise} using random function and is added to the original dataset using the equation *(4)* to create the noisy dataset array *DA* with the number of noisy factors as *nf*.

$$DA = set(D + \lim_{0 \to nf} f_{noise}) \quad \textbf{(4)}$$

2.3 Machine Learning Algorithms

2.3.1 *Regression Algorithms:* The regression algorithms chosen for this work are linear regression and polynomial regression [11]. L2/Ridge regularization is applied to both the linear and polynomial regression with the alpha value chosen using cross-validation. The regression algorithms are implemented in Python 3.6 [6] using the Scikit-learn framework [7] due to the ease of use, fine grained control, huge community base and easy integration with the visualization framework MatplotLib [12]. The bike dataset is used for regression algorithms.

2.3.2 *Classification Algorithm:* The classification algorithm used in this study is random forest. The implementation and frameworks used are the same as explained in section 2.3.1. The number of estimators is chosen using cross-validation. The wine dataset and mobile pricing dataset are used for classification algorithm.

2.4 Evaluation

Table 7 and Table 8 shows the metrics that are used for analyzing the performance of regression algorithms and classification algorithms respectively.

Table 7: Metrics Used for Evaluation of Regression Algorithms

Name of the Metric	Abbreviation	Rationale of Usage

Root Mean Squared Error	RMSE	A preferred metrics as it is very sensitive to outliers and gives a better perspective about the effect of noise in the overall model [13].
R Squared Score	R2	Preferred as it is easily computable, it significantly punishes and highlights the far fitted predicted values.

Table 8: Metrics Used for Evaluation of Classification Algorithm

Name of the Metric	Abbreviation	Rationale of Usage
F1 score	F1	It consolidates the precession and recall metrics into one number.
Accuracy		It provides a quantifiable metric for the confusion matrix.

Two sets of 15 noisy datasets – 10 with sample noise and 5 with feature noise are created for each of the three datasets as per the methodology explained in section 2.2 .

For each of the dataset, ten percent of the data is reserved as test set and ninety percent of the data is used for training and cross validation. For sample noise, noisy datasets are generated by substituting the values of $k=10$, n ranging from 10 to 100 incremented by k, and $s=20$ in equation (1), (2) and (3) . Hence the attribute noise is added in steps of ten per cent from ten percent of the data size (Example Figure 11) up to hundred percent with a noise spread spectrum (s) of 20% (Example Figure 12)

For datasets with feature noise, features are added in gradient of one and using equation (4) with n ranging from 1 to 5. 90% of the data was used for k-fold cross validation (number of folds=10) and training and 10% was reserved as out-of-sample data for testing the final model.

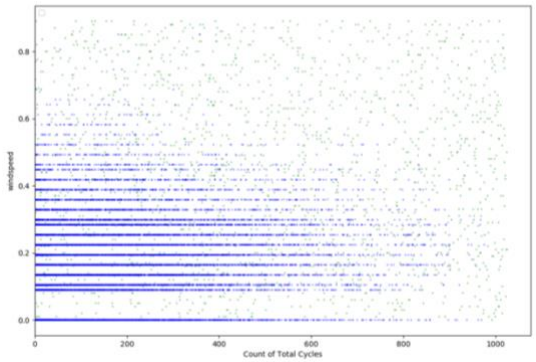


Figure 11: Ten percent noise added to the feature windspeed of the bike dataset where the blue dots are real data and green dots are noise

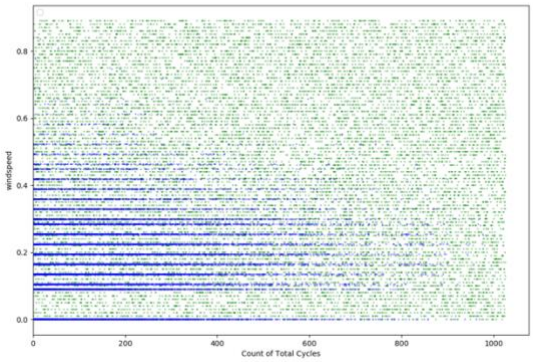


Figure 12: Hundred percent noise added to the feature windspeed of the bike dataset where the blue dots are real data and green dots are noise

The metrics mentioned in

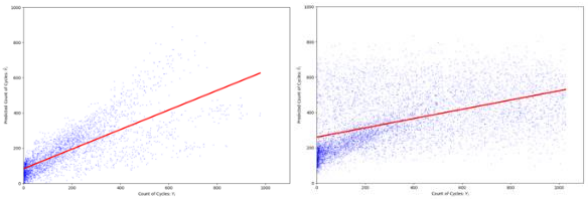
Table 7 are computed for each dataset using the out-of-sample 10% test data and results are plotted to find the strength of the impact of noise on regression algorithms (linear and polynomial regression with degree three[1]) and the metrics in Table 8 for the classification algorithm (random forest). For both types of algorithms, the results are computed with a 95% confidence level.

3 RESULTS & DISCUSSION

3.1 Regression Algorithms

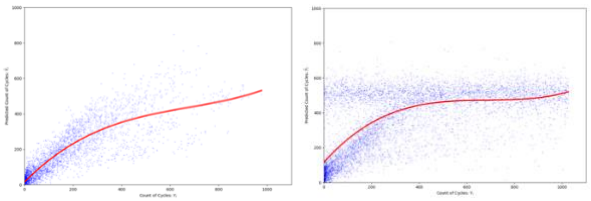
Figure 13 shows the linear fit on clean data and the fit on data with 100% sample noise. Figure 14 shows the polynomial fit on clean data and the fit on data with 100% sample noise. From both the figures, it is evident that the model fits to the noise reducing performance.

[1] Degree three for polynomial regression based on comparing the results for degrees 1 to 5.



(a) (b)

Figure 13: (a) Linear Fit With Clean Data (b) Linear Fit With 100% Noisy Data



(a) (b)

Figure 14: (a) Polynomial Fit With Clean Data (b) Polynomial Fit With 100% Noisy Data

The base line with mean prediction has a R2 score for 0.01 (+/- 0.02) with 95% confidence level on zero noise data. The linear model with zero noise yields a cross validated R2 with 95% confidence interval as 0.57 (+/- 0.04) and for the polynomial model it is 0.68 (+/- 0.03). With increase in sample noise by 100%, both linear regression and polynomial regression show a similar trend with a 32% decrease in R2 with values 0.28 (+/- 0.02) & 0.39 (+/- 0.02). They show a steady decrease with each 10% addition of sample noise from zero-noise (Figure 15).

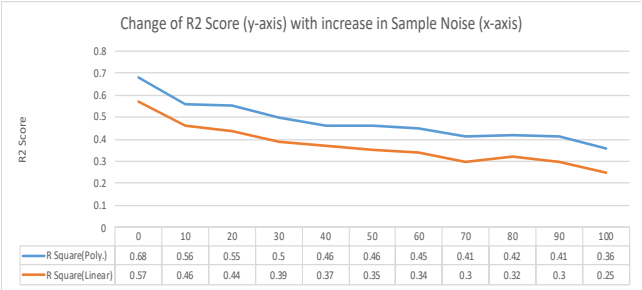


Figure 15: Change of R2 with increase in noise for tuple noise in Linear Regression and Polynomial Regression

Similarly, in terms of RMSE both linear and polynomial regression show similar trend with increase in sample noise. The RMSE increases steadily up to 20% increase in noise and then onwards there is no steady increase. With 100% increase in noise, polynomial regression suffers an RMSE increase of 61.7% whereas linear regression suffers from 49.7% RMSE increment. The graphical representation of the RMSE trend is given in Figure 16.

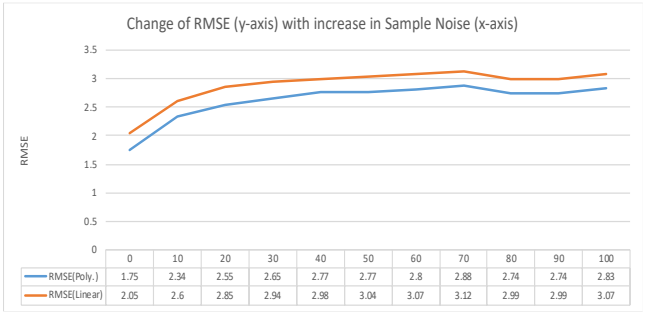


Figure 16: Change of RMSE with increase in noise for tuple noise in Linear Regression and Polynomial Regression

In terms feature noise, surprisingly, both linear and polynomial regression shows no/negligible change of performance as shown in the Figure 17 and Figure 18.

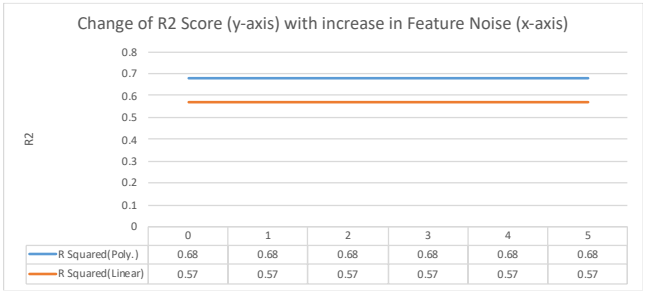


Figure 17: Change of R2 with increase in feature noise for Linear Regression and Polynomial Regression

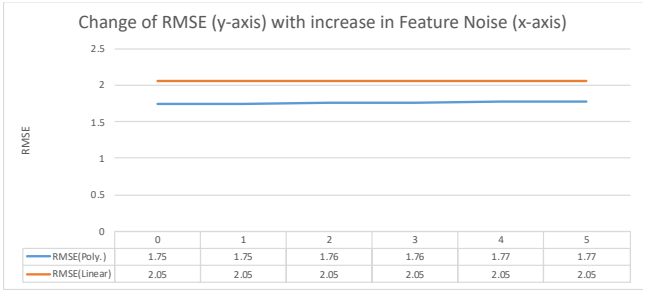


Figure 18: Change of RMSE with increase in feature noise for Linear Regression and Polynomial Regression

3.2 Classification Algorithm

From the experiments, the random forest classification algorithm shows steady deterioration in both accuracy and F1 score with increase in sample noise. The baseline model with random prediction has an accuracy of 50% and the random forest model with zero noise yields an accuracy of 93%. With 10% increase in noise, there is a mean deterioration of 6% of Accuracy and 2.5% of F1 score for the two datasets considered. For 100% increase in sample noise, the Accuracy decreases by an average of 26% and F1 score decreases by an average of 23.5% across both the datasets as shown in Figure 19, Figure 20.

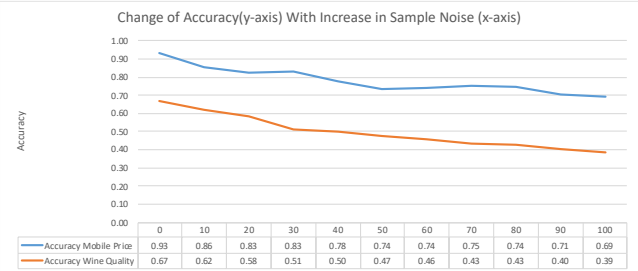


Figure 19: Change of Accuracy with Increase in Sample Noise for Random Forest

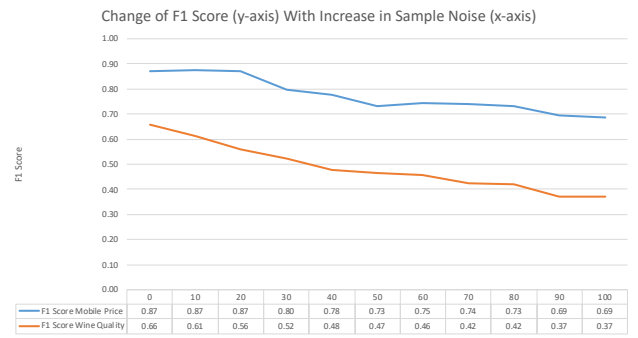


Figure 20: Change of F1 Score with Increase in Sample Noise for Random Forest

3.3 Strength of Impact of Noise

From the results in section 3.1 and 3.2, it can be concluded that the average strength of the impact of sample noise per 10% increase in noise on linear regression and polynomial regression is 3.2% (in terms of R2) decrement and that of feature noise is negligible. For random forest the strength of impact of sample noise is 3% in the negative direction per 10% increase of noise (in terms of accuracy). The results confirm the observations in the related works [1] and [2] with exception for feature noise.

LIMITATIONS & OUTLOOK

In this work, the hypothesis of degradation of performance with increase in noise with regression and classification algorithms is studied. The study can be extended on other types of machine learning algorithms such as clustering and neural networks.

Further, this works only considers the strength of impact of noise on accuracy metrics. The strength of impact of noise on other parameters such as size of the model and the time taken for learning could also be studied.

ACKNOWLEDGEMENTS

This analysis was conducted as part of the 2018/19 Machine Learning Module CS7CS4/CS4404 at Trinity College Dublin, University of Dublin, Dublin, Ireland [14].

REFERENCES

[1] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial intelligence review*, vol. 22, no. 3, pp. 177-210, 2004.
[2] 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," *Artificial intelligence review*, vol. 33, no. 4, pp. 275-306, 2010.
[3] H. Fanaee-T, "Bike Sharing Dataset," C. B. System, Ed., ed. UCI Machine Learning Repository: Laboratory of Artificial Intelligence and Decision Support (LIAAD), University of Porto, 2013.
[4] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Wine Quality Dataset ", vinhoverde.pt, Ed., ed. kaggle.com, 2009.
[5] S. Abhishek, "Mobile Price Classification," V. Kaggle.com, Ed., ed. Kaggle: Kaggle.com, 2017.
[6] (2016). *Python 3.6* 3.6, Python Software Foundation. [Online]. Available.
[7] (2018). *scikit-learn* 0.20.0, scilit-learn.org. [Online]. Available.
[8] S. Lovish, C. Viren, C. Debrup, B. R. Aneek and T. S. Arun. (2018). *Academic Research*, github.com. [Online]. Available: https://github.com/ats0stv/AcademicResearch.
[9] *Robust Scaler* [Online]. Available: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#robustscaler.
[10] C.-M. Teng, "Correcting Noisy Data," in *ICML*, 1999: Citeseer, pp. 239-248.
[11] J. P. Mueller and L. Massaron, *Machine learning for dummies*. John Wiley & Sons, 2016.
[12] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering*, *IEEE COMPUTER SOC*, vol. 3, 9, pp. 90-95, 2007.
[13] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247-1250, 2014.
[14] J. Beel and *D.* Leith "*Machine Learning (CS7CS4/CS4404)*," ed. *Trinity College Dublin, School of Computer Science and Statistics*, 2018.