Study the effect of training set noise over different regression algorithms and settings

Alevizopoulou Sofia, AM: 2022201704002; Avgeros Giannis, AM: 2022201704003

Tsiatsios Georgios, AM: 2022201704023, Lefteris Balteas AM: 2022201704013

Athens 2019, MSC Data Science, Applied Data Science

Abstract

It is very common in machine learning techniques to deal with noisy data, which may affect the accuracy of the resulting data models. For that reason, the handling of noise is a key aspect in machine learning, critical for obtaining reliable models from data. In this assignment, we address this issue by comparing how the noise affects Linear, Lasso and Ridge regression of 3 different datasets. We will compare the final outcomes from the original datasets with the outcomes from the noisy dataset using the root mean squared error as metric. We saw that, regardless of the feature we chose to alter, the RMSE is getting bigger and bigger and the "fit" of the regression line is getting worst. We also noticed that noise in target feature affects more the predicted values than the noise at the other features of the dataset.

Keywords

Attribute noise, noisy data, regression algorithms, t- test

1. Introduction

Machine learning techniques, and specifically supervised learning techniques, are applied to extract novel and interesting information from data collected out of real-world problems. An important characteristic of these datasets is that the data frequently contains noise. Noisy data may be the consequence of human error due to mistakes in the translation phase or due errors while collecting them. Noisy data, may produce bias to the learning process, and as a result, is more difficult for learning algorithms to form accurate models from them. Developing learning techniques that effectively deal with noisy data is a key aspect in machine learning.

In this assignment, we provide a comparison on the effect of attribute noise and class noise on three regression models (a) Linear Regression, (b)Lasso Regression, (c) Ridge Regression. We are going to use 3 different datasets from different domains, each of them describes a different regression problem. The first dataset is the *air Qualit*y dataset that contains instances of hourly averaged responses from metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. [10] The second dataset is the *Computer Hardware Data Set* That contains some characteristics like cache memory, machine cycle time, etc for different computer models. [11] The third dataset is *Facebook metrics Data Set,* data is related to post's published during the year of 2014 on the Facebook's page of a renowned cosmetics brand. [12] We used these datasets to generate the "noisy" version of them. Finally, having the clean and the noisy version of each dataset, we compared the effect of noise on the models created (9 different models, for each dataset 2 different kind of noise).

Training a linear regression model is the process of finding out the most suitable coefficients for the linear function that best describes the input variables. On each step, the algorithm seeks to eliminate the error produced by the predictions and the real values. It does so by constantly seeking for a way to limit the value of a function (the so-called loss function) that helps us measure the error. In regression problems, we need evaluation metrics designed for comparing continuous values. Root Mean Square Error (RMSE) is the most commonly used regression loss function. RMSE is the sum of squared distances between our target variable and predicted values. We would like to minimize the RMSE function as much as possible, so the prediction will be as close as possible to the ground truth by updating the coefficients.

We are going to use the statistical test, two paired t-test to consider the effect of noisy data at the outcomes of the regression models. The results of statistical tests enable us to highlight the characteristics of the different approaches, clean and noisy data. The whole analysis is developed from the following null (initial) hypothesis.

Initial hypothesis: Given the characteristics of each learning technique, we initially propose that noisy training data don't affect the outcome of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression

The structure of the present paper is as follows: in Sect. 2 summarization related work in noise analysis. In Sect. 3 presentation of the three Regression algorithms and the learning techniques. Sect. 4 description of the data sets, and explanation of the methodology used for the noise generation. Sect. 5 presents and analyzes the results of the different experiments. Sect.6 describe and run hypothesis tests based on null hypothesis. Sect. 7 describes the source code. Sect. 8 summarization the overall conclusions.

2. Related work

Understanding the impact of noisy data in the performance of machine learning algorithms is a key issue for improving algorithms reliability. Below, we will describe some techniques for noise generation. Noise can be added only in the target feature, in all features or in a combination of them. [1] In general, label noise affects significantly the model. [7] In current researches, some machine learning techniques are considered more "robust" to noise, errors and missing values than others. [1] Except from that, there also many alternative multivariate linear regression methods designed to face the various data uncertainties [8], whereas other approaches like Lasso Regression known as `1-penalized regression" are not equipped to deal with noisy or missing data. The idea of a statistical test and more specifically the concept of P-value as a measure of the likelihood of the observed value of the statistic under the "null hypothesis" has been introduced [16] in many researches. This kind of statistical test will also be used at this case.

2.1. Noise Generation

Noise generation can affect multiple aspects of the dataset. We summarize them and present the following bullets of the various aspects of noise:

- Noise can be introduced in the input attributes or/and the target feature of the data
- Noise follows a specific distribution, for example, normal or Gaussian
- Noise have values that can be relative to min, max, std (standard deviation) of each variable or to the variable value itself.

We have implemented one python script which generates 2 kinds of noisy datasets (noisy features and noisy target features) based on the input dataset. The process is described below.

2.1.1. Attribute noise

This noise category adds noisy data at a specific feature of the train dataset. As we want to study the effect of noise when trained with a clean dataset versus being trained with noisy train dataset, we will not alter test dataset.

Firstly, we define the percentage of noise which could be between 0%-100%, let that be the Fc variable. After using this percentage, we find out how many random variables will be generated by multiplying the percentage of noise Fc with the number of instances, so we have N random values. Now, for each feature, we generate N random numbers Rv with a Gaussian distribution and within a range between the max and min of the corresponding feature. Then we generate another random set of ids with a uniform distribution within the range 1 to N, which produces the instance ids whose data value should be overwritten by the generated noisy value. [1]

Specifically, we create a dictionary of lists. Each element of the dictionary is of the type {Feature name: list}, where each list contains N (number of instances multiplied by the fraction of noise) elements. For every such element of a feature list we generate two values:

i) Noisy value: Value that will be part of a gaussian distribution, with min and max the feature's min max value respectively, and mean value the mean value of the feature. Furthermore, we define also standard deviation to be (max-min)/6.
ii) Id: Each feature noisy value will have a corresponding id of a record to be altered. The id's are randomly selected using a uniform distribution that will generate unique (and not duplicate id's)

Using (i) and (ii), we assign to the record with id the (ii) id and we replace for the specific feature the noisy (i) feature value.

2.1.2. Target feature noise

This kind of noise adds noise at the target feature of the train dataset. The process is the same as before, as we seek to generate noise values for one feature instead of the multiple features we did on the previous feature case. Thus, we generate noisy values and ids for the target feature only.

2.1.4. Level of noise

We have added 3 different level of noise: 15%, 35%, 50%. As a result, 6 noisy datasets for each different dataset will be generated.

2.2 Stratified split for test train dataset

We wanted to create two different datasets for the implementation of noise effect. The first dataset would be the train one and the second one would be the test one. The test dataset will not contain any noise as stated previously. For the train dataset, we create two different versions of it, one with noise for the attribute and one with noise for the class dataset.

What is important here is the fact that we needed to perform the split in a stratified way. Thus, we wanted for the test and the train distribution to be as similar as possible. Implementation was done following the principles stated below:

1. We sort the dataset based on the target feature
2. We split it the sorted dataset to lists of 10 items.
3. The proportion of split to test and train is the number of items we select out of the 10 items. We decided to split the dataset based on 80 % train – 20% test, so

the number of elements taken from the sorted 10 items would be 2 items. Those 2 items would be added to the test dataset, whereas the remaining 8 are part of the training dataset.

As we see by that, we force with the way we create the datasets that the distribution of the train dataset would be preserved and be present also to the test dataset.

3. Regression algorithms

3.1 Regression techniques

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent and independent variable. This technique is used for forecasting, time series modelling and finding the curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.

One of the first steps in a regression analysis is to determine if multicollinearity is a problem. Multicollinearity, or collinearity, is the existence of near-linear relationships among the independent variables. Multicollinearity causes two basic types of problems. The coefficient estimates can swing rapidly based on which other independent variables are in the model, so coefficients become very sensitive to small changes in the model. Multicollinearity reduces the precision of the estimate coefficients, which makes the statistical power of the regression model weak.

In regression problems, we need evaluation metrics designed for comparing continuous values. We would like to minimize the MSE (loss function) for a specific data-point as much as possible so the prediction will be as close as possible to the ground truth. This is done by learning the parameters, executing an iterative process that updates coefficients at every step and reduce the loss function as much as possible until the minimum point of the loss function has been reached.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2$$

Where n: number of instances, Y^: predicted values, Y: observed values

3.2. Linear regression

Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line. The dependent variable is continuous, independent variable(s) can be continuous or discrete, and the regression line is linear.

Its equation is $Y = a+b1X_1+b2X_2+b3X_{3+\ldots}$

where Y: the response, a: the intercept, b: model coefficients, *Xn* (the nth feature)

This equation can be used to predict the value of target variable based on given predictor variable(s). We would like to minimize the MSE (loss function) for a specific data-point as much as possible. Linear Regression is very sensitive to Outliers. Having noisy data can influence on the computation of MSE and as a result the regression line. Prediction errors can occur due to two sub components or combination of them. First component is the bias and the second is the variance (multicollinearity problem). [2]

3.3. Ridge Regression

Ridge regression adds "squared magnitude" of coefficient as penalty term to the loss function, it is known as L2 regularization. It is another regression method and used when the data suffers from multicollinearity, independent variables are highly correlated. In multicollinearity, even though the least squares estimates are unbiased, their variances are large. Large variances mean that the observed value is far from the true value. Adding a degree of bias, standard errors are reduced. Ridge regression solves the multicollinearity problem through shrinkage parameter λ (lambda). Ridge regression is a well-known approach to dealing with noisy data. [4]

Its equation is $Y=a+b*X+\varepsilon$
where Y: the response, a: the intercept, b: model coefficients, *Xn* (the nth feature), ε: the error. Error term is the value needed to correct the prediction error between the observed and predicted value. The highlighted part below represents L2 regularization element. [6]

$$\sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2 + \boxed{\lambda \sum_{j-1}^{m} \hat{\beta}_j}$$

If *lambda* is very large then it will add too much weight and it will lead to under-fitting, if *lambda* equals to zero then we have linear regression. We can understand that it's important how *lambda* is chosen. Minimizing the sum of the squares of the vertical deviations from each data point to the line enforce β coefficients to be lower but it does not enforce them to be zero.

3.4. Lasso Regression

Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function, its known as L1 regularization. It penalizes the absolute size of the regression coefficients, reduces the variability and improving the accuracy of linear regression models.

Lasso regression differs from ridge regression in a way that it uses absolute values in the penalty function, instead of squares. This leads to penalizing values which causes some of the parameter estimates to turn out exactly zero. Larger the penalty applied, further the estimates get shrunk towards absolute zero. Therefore, you might end up with fewer features included in the model than you started with, which is a huge advantage. Lasso regressor is not equipped to deal with noisy or missing data. [2]

Its equation is Y=a+b*X+ε

where Y: the response, a: the intercept, b: model coefficients, *Xn* (the nth feature), ε: the error. Error term is the value needed to correct the prediction error between the observed and predicted value. The highlighted part below represents L1 regularization element. [6]

$$\sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2 + \boxed{\lambda\sum_{j=1}^{m}|\hat{\beta}_j|}$$

If *lambda* is a large value will make coefficients zero hence it will under-fit, if *lambda* equals to zero then we have linear regression. The key difference between Lasso and Ridge techniques is that Lasso shrinks the less important feature's coefficient to zero thus, removing some feature altogether. So, this is effective for feature selection in case we have a very big number of features.

4. Design of the dataset

In our experiments we will use the 3 different datasets. All of them are based on real data (data descriptions of real objects or events), may contain a certain amount of background noise, erroneous values, and so on. All the data sets have as target numerical attributes. Noise will be added only at the train data. Before adding noise, we have splitted the dataset into train and test.

4.1. Air-Quality

The air quality dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device and 15 features. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year) representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. At this case, we try to find the best regression model at which the predict the amount of C6H6(GT) will be predicted. There is a readme file available at which there are detailed information about the dataset. [10]

As said before, this is a real dataset so for sure it contains noisy data. Because we don't know exactly the percentage of noise we generated other datasets based on this at which we will include several noise levels on the features and on the target feature. The generated datasets contain noise data at the target class – C6H6(GT) - whereas the other noisy datasets contains noise data at all features of the dataset except the target one. The generation of noise has been analyzed on Sect. 2.1.

4.2. Computer Hardware

The computer hardware dataset contains 209 instances and 10 features which describe the characteristic of a computer: model name, cache memory, machine cycle, published performance, etc. At this case, we try to find the best regression model at which the performance of the computer will be predicted. There is a readme file available at which there are detailed information about the dataset. [11] The process of noise generation is the same with the *Air-quality dataset.*

4.3. Facebook metrics

The Facebook metrics dataset contains 500 instances and 19 attributes. Some of them are known prior to post publication and the 12 remaining features used to evaluate the post impact. The data is related to posts' published during the year of 2014 on the Facebook's page of a renowned cosmetics brand. At this case, we try to find the best regression model at which the -Total Interactions -with a Facebook page will be predicted. There is a readme file available at which there are detailed information about the dataset. [12] The process of noise generation is the same with the *Air-quality dataset.*

5. Results of the experiments

Test dataset is the same for all different cases. We have run all regression problems for 6 different noisy data.

x_f_train_noisy.xlsx→ x is the percentage of noise at all the features of the dataset

x_c_train_noisy.xlsx→ x is the percentage noise at the target feature of the dataset

5.1 Air-quality

Regarding the air-quality we run 3 regression algorithms for different noise levels and types of noise which are listed below. We have used the root mean squared error as metric. We can see that using noisy data for training the model we result in bigger RMSE, so the "fit" of the regression line is getting worst.

Linear

Table 1: RMSE Linear Regression

Datasets	RMSE
clean data:	1.1389
0.15_c_train_noisy.xlsx	19.9225
0.15_f_train_noisy.xlsx	12.4318
0.35_c_train_noisy.xlsx	47.5515
035_f_train_noisy.xlsx	22.5007
0.50_c_train_noisy.xlsx	68.4762
0.50_f_train_noisy.xlsx	28.1405

Lasso

Table 2:RMSE Lasso Regression

Datasets	RMSE
clean data:	1.1908
0.15_c_train_noisy.xlsx	19.9185
0.15_f_train_noisy.xlsx	12.4300
0.35_c_train_noisy.xlsx	47.5579
035_f_train_noisy.xlsx	22.4879
0.50_c_train_noisy.xlsx	68.4858
0.50_f_train_noisy.xlsx	28.1286

Ridge

Table 3:RMSE Ridge Regression

Datasets	RMSE
clean data:	1.1389
0.15_c_train_noisy.xlsx	19.9225
0.15_f_train_noisy.xlsx	12.4318
0.35_c_train_noisy.xlsx	47.5515
035_f_train_noisy.xlsx	22.5007
0.50_c_train_noisy.xlsx	68.4762
0.50_f_train_noisy.xlsx	28.1405

From the next plots, Figure 1, Figure 2 we can see that adding several levels of noise, RMSE is getting bigger.

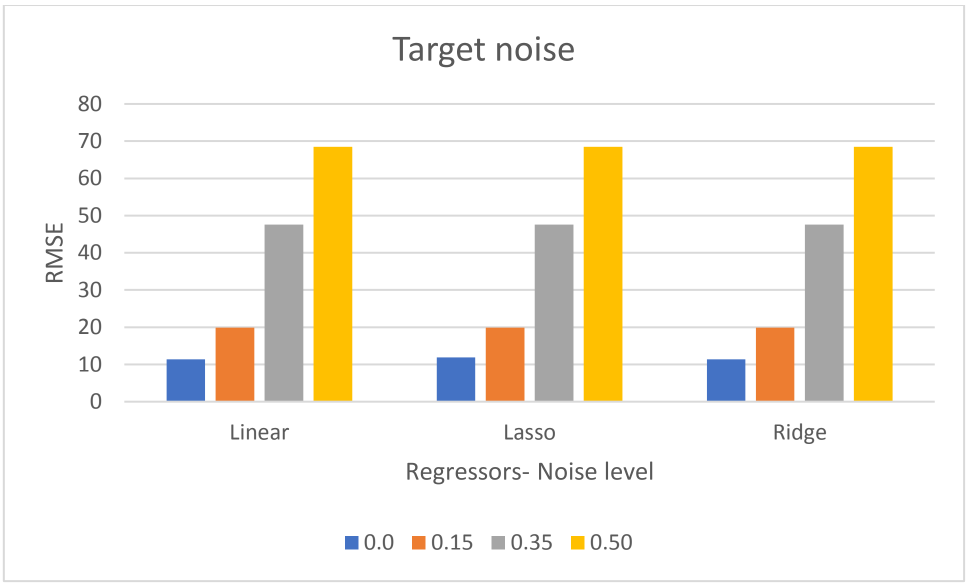

Figure 1:RMSE when target noise added

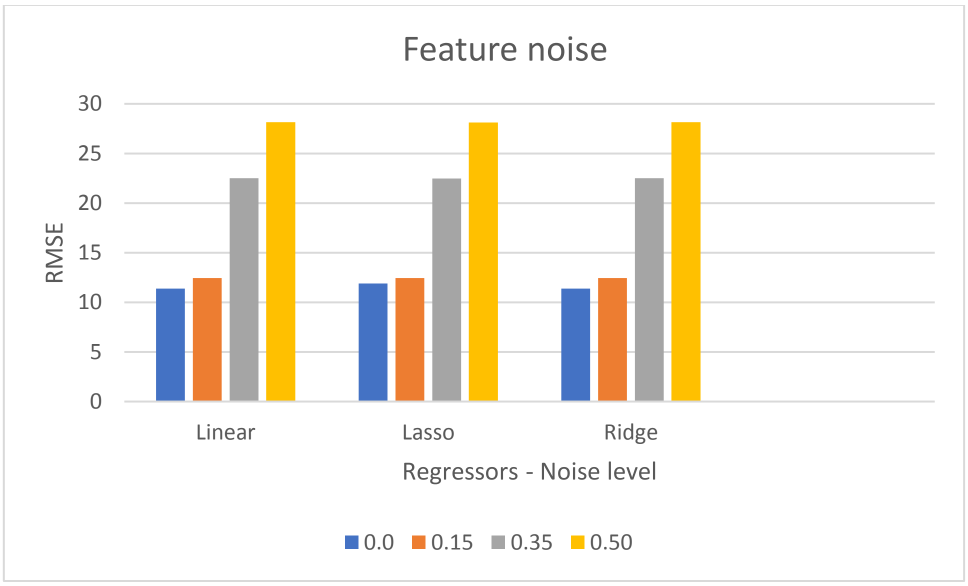

Figure 2:RMSE when feature noise added

Also, we can see that target noise affects more the predictions of the model than the feature noise. Where the noise level is big (> 35%) RMSE of the dataset with noisy target feature is two times the RMSE of the datasets with noisy features and clean the target feature, see Figure 3.

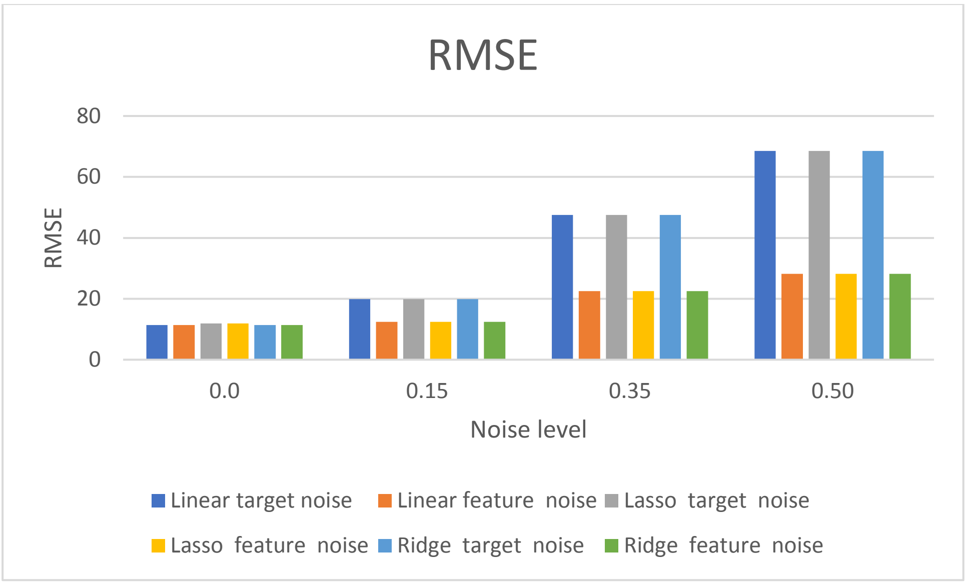

Figure 3: RMSE with target - features noise

5.1. Computer Hardware

We can notice the same behavior on computer hardware dataset. Running 3 regression algorithms for different noise levels and types of noise and using the root mean squared error as metric, we can see that using noisy data for training the model we result in bigger RMSE, so the "fit" of the regression line is getting worst.

Linear

Table 4: RMSE Linear Regression

Datasets	**RMSE**
clean data:	38.5487
0.15_c_train_noisy.xlsx	93.7774
0.15_f_train_noisy.xlsx	65.2908
0.35_c_train_noisy.xlsx	212.6189
035_f_train_noisy.xlsx	110.6693
0.50_c_train_noisy.xlsx	280.8337
0.50_f_train_noisy.xlsx	147.3978

Lasso

Table 5:RMSE Lasso Regression

Datasets	**RMSE**
clean data:	38.5239
0.15_c_train_noisy.xlsx	93.7374
0.15_f_train_noisy.xlsx	65.2561
0.35_c_train_noisy.xlsx	212.6021
035_f_train_noisy.xlsx	110.6317
0.50_c_train_noisy.xlsx	280.7999
0.50_f_train_noisy.xlsx	147.3213

Ridge

Table 6: RMSE Ridge Regression

Datasets	**RMSE**
clean data:	38.5486
0.15_c_train_noisy.xlsx	93.7772
0.15_f_train_noisy.xlsx	65.2904
0.35_c_train_noisy.xlsx	212.6189
035_f_train_noisy.xlsx	110.6692
0.50_c_train_noisy.xlsx	280.8332
0.50_f_train_noisy.xlsx	147.3973

The above results Table 4, Table 5, Table 6, are displayed also at the below diagrams. It is obvious that adding noise, RMSE is getting bigger.

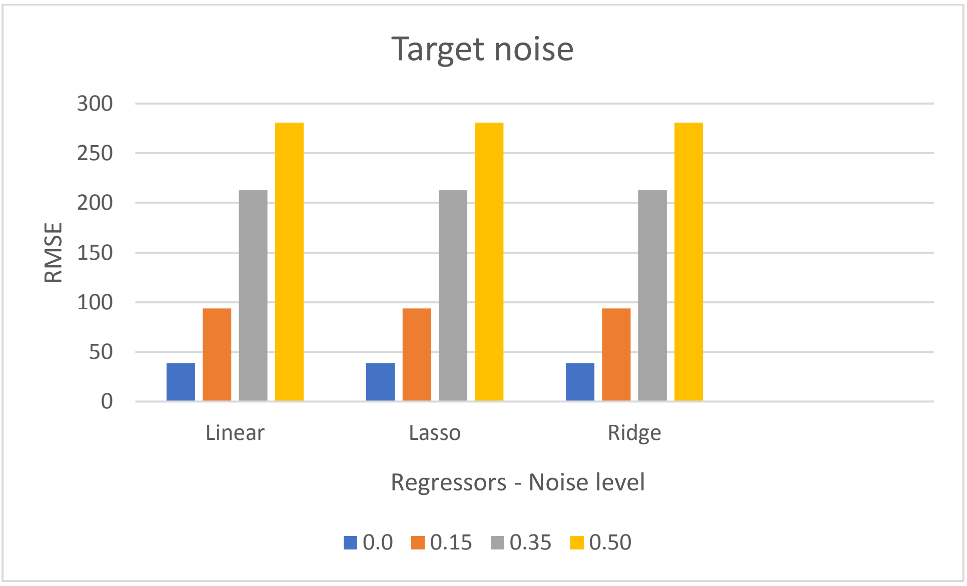

Figure 4: RMSE when target noise added

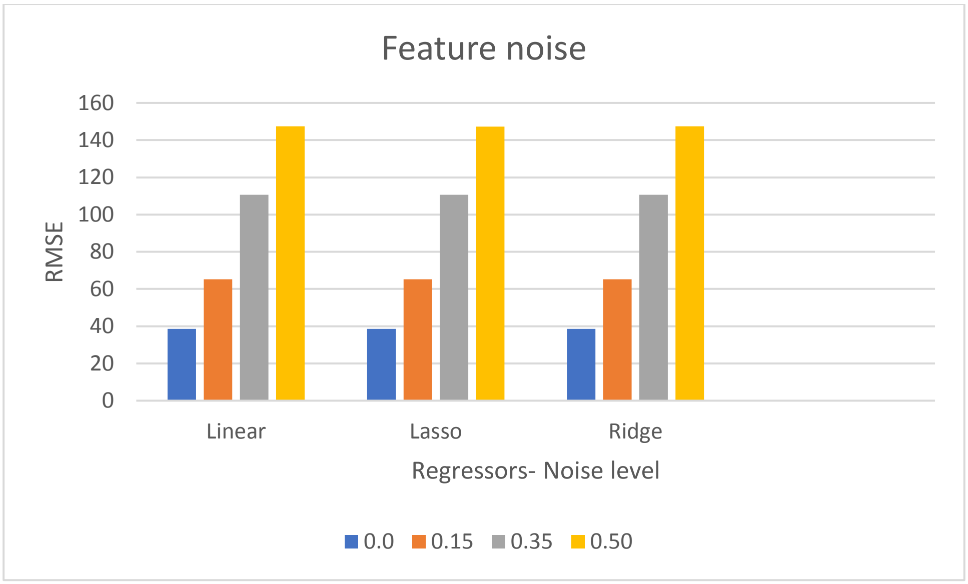

Figure 5: RMSE when feature noise added

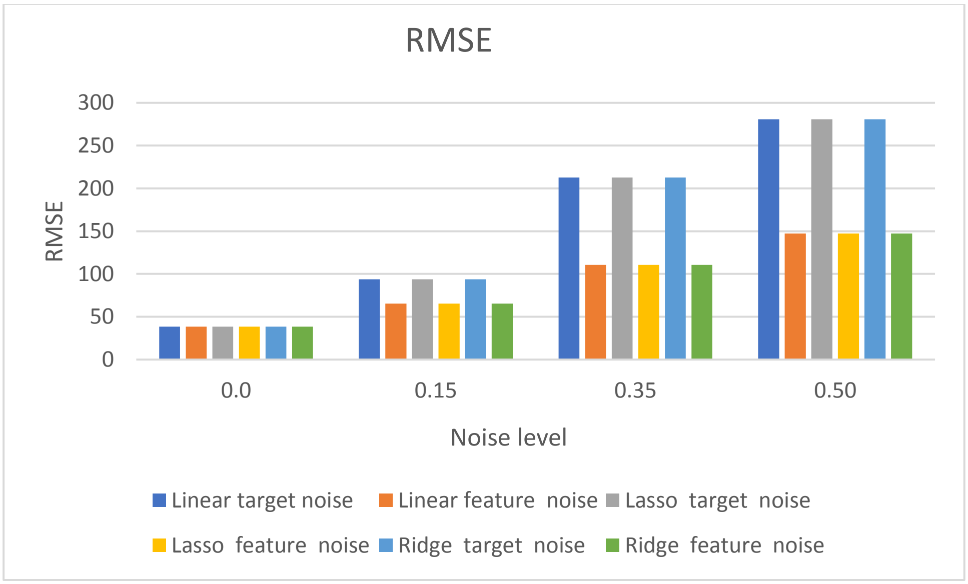

Figure 6: RMSE with target - features noise

5.2. Facebook Metrics

The same behavior is noticed on Facebook metrics dataset, also. Running 3 regression algorithms for different noise levels and types of noise and using the root mean squared error as metric, we can see that using noisy data for training the model we result in bigger RMSE, so the "fit" of the regression line is getting worst.

Linear

Table 7: RMSE Linear Regression

Datasets	RMSE
clean data:	0.0000
0.15_c_train_noisy.xlsx	498.5506
0.15_f_train_noisy.xlsx	247.5395
0.35_c_train_noisy.xlsx	1183.3034
035_f_train_noisy.xlsx	314.5667
0.50_c_train_noisy.xlsx	1712.6426
0.50_f_train_noisy.xlsx	379.1050

Lasso

Table 8: RMSE Lasso Regression

Datasets	RMSE
clean data:	5.7487
0.15_c_train_noisy.xlsx	499.0478
0.15_f_train_noisy.xlsx	247.5375
0.35_c_train_noisy.xlsx	1156.2907
035_f_train_noisy.xlsx	314.5598
0.50_c_train_noisy.xlsx	1694.8640
0.50_f_train_noisy.xlsx	379.0970

Ridge

Table 9: RMSE Ridge Regression

Datasets	RMSE
clean data:	0.0002
0.15_c_train_noisy.xlsx	498.5505
0.15_f_train_noisy.xlsx	247.5395
0.35_c_train_noisy.xlsx	1183.3022
035_f_train_noisy.xlsx	314.5667
0.50_c_train_noisy.xlsx	1712.6423
0.50_f_train_noisy.xlsx	379.1050

All these results are also displayed at the next plots, where adding noise the RMSE is getting bigger and far away from the best fit.

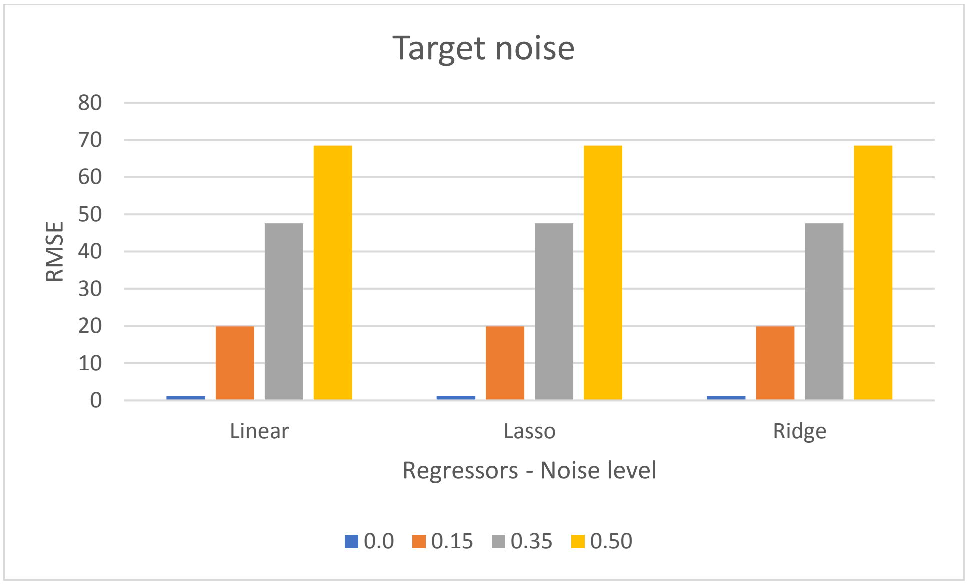

Figure 7: RMSE when target noise added

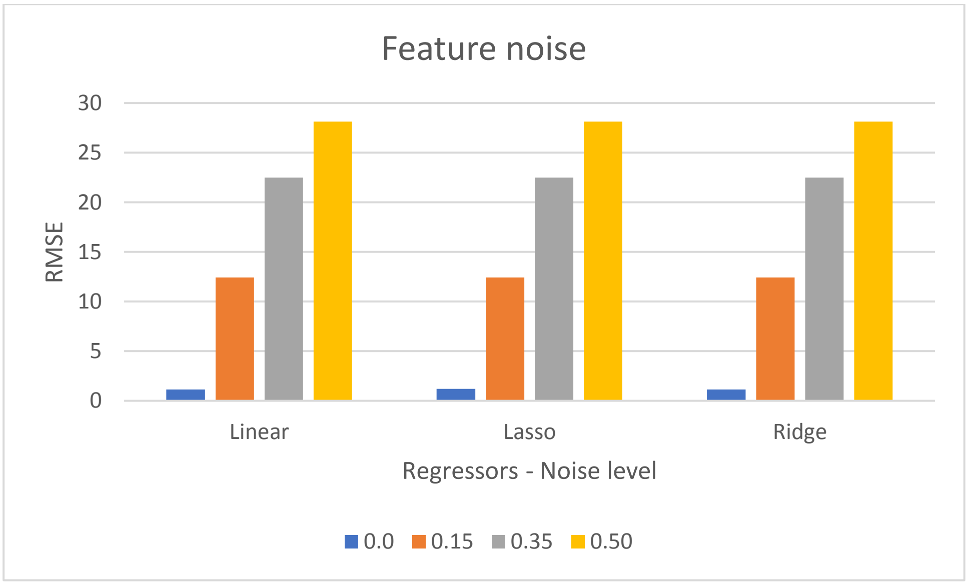

Figure 8: RMSE when feature noise added

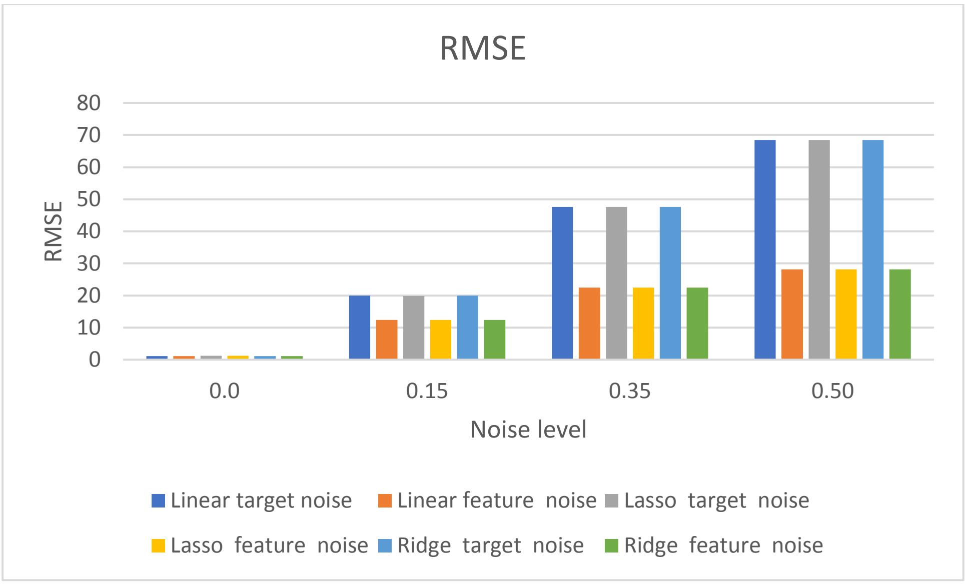

Figure 9: RMSE with target - features noise

6. Hypothesis testing

6.1. Null hypothesis

Hypothesis testing is a way to test the results of a survey or experiment. As first step the null hypothesis should be defined. The null hypothesis is a general statement or default position that there is not, or there is a relationship between some measured phenomena. Hypothesis testing using some statistical tests which will reject or no the null hypothesis. If the null hypothesis will be rejected, then the initial hypothesis we have done is fault and we accept the alternative hypothesis. We have defined three different null hypotheses that aim to respond at some main questions on the dataset.

The questions and null hypotheses are listed below:

1. *Question*: Does the noisy data affect the final outcomes of regression models?

   Null hypothesis – H0: Given the characteristics of each learning technique, we initially propose that noisy training data, don't affect negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 clean datasets and the noisy versions of them.

   Alternative hypothesis – H1: Given the characteristics of each learning technique, noisy training data, does affect the final prediction of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 datasets and the noisy versions of them.

2. *Question*: Does different noise types affect the prediction of regression models? We have two noise types: feature noise and target noise and we used the same noise level.

   Null hypothesis – H00: Given the characteristics of each learning technique, we initially propose that feature noise and target noise in training data, affect equally the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 2 different noise levels of each dataset.

   Alternative hypothesis – H01: Given the characteristics of each learning technique, the feature noise and target noise in training data, don't affect equally the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 2 different noise levels of each dataset.

3. *Question*: Does the noise level affect the final prediction of regression models?

   Null hypothesis – H000: Given the characteristics of each learning technique, we initially propose that with higher proportion of noise in training data, doesn't affect negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 clean datasets and the noisy versions of them.

   Alternative hypothesis – H001: Given the characteristics of each learning technique, the high level of noise in training data, affects negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 datasets and the noisy versions of them.

6.2. Two sample t-test

As mentioned before, we are interested in comparing the performance of the different learning techniques on data with different proportions of attribute noise and class feature noise in the training data set, using the same test dataset. For that scope we will use the Independent t-test.

T-test is used to determine if two population means are equal or not, the most appropriate version of it is the paired t-test. It is about doing two different tests on same dataset (the test dataset is the same, we just change the train dataset).

With a two-sample t test, we are comparing the means for two different samples. Paired t-test is used when there are two measurements on the same item. T score is a ratio on how different two groups are. The larger the t score, the largest the difference between groups. Each t-value has a p-value to go with it. A p-value is the probability that the results from your sample data occurred by chance. P-values are from 0% to 100%. In most cases, a p-value of 0.05 (5%) is accepted, proving that the initial hypothesis is rejected, statistically excluding the spontaneous occurrence of the event .

6.3. Running the hypothesis tests

We calculated the statistical tests, t-test using the *ttest* from *spicy*. In hindsight, each time we train 2 different regression models, and for each prediction we calculate each ones RMSE against the true value. After, we use these values (list of predicted values) as an input to the t-test.

The null hypotheses will be rejected if the calculated p-value is less than the significance level of 0.05. If p-value is less than 5%, then the test concludes that there is a statistically significant difference between the two populations. In any other case, there is no statistically significant difference between the two populations and the test fails to reject the null hypothesis.

6.3.1. Hypothesis testing for Null hypothesis – H0

We run H0[1] hypothesis testing between the predicted values from a model which has been trained with clean data and the predicted values from a model which has been trained with noisy data.

In the following tables, we display the results from t-test. In all different cases the p-value is significantly less than 0.05 which is the significance level and as a result the null hypothesis is rejected. Different means result in different predicted values between the clean and the noisy datasets.

[1] *Null hypothesis – H0:* Given the characteristics of each learning technique, we initially propose that noisy training data, don't affect negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 clean datasets and the noisy versions of them.

6.3.1.1. Air-quality

At the next tables are displayed the results from t-test.

Linear

Table 10: T-test Linear Regression H0 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx
-
Clean dataset
	t = -15.03
	p = 2.42e-49
		Rejected

0.15_f_train_noisy.xlsx
-
Clean dataset
	t =7.22
	p = 1.18e-12
		Rejected

0.35_c_train_noisy.xlsx
-
Clean dataset
	t = -39.54
	p = 8.26e-286
		Rejected

035_f_train_noisy.xlsx
-
Clean dataset
	t = 14.68
	p = 3.25e-47
		Rejected

0.50_c_train_noisy.xlsx
-
Clean dataset
	t =- 61.20
	p = 0.0
		Rejected

0.50_f_train_noisy.xlsx
-
Clean dataset
	t =18.98
	p = 1.55e-76
		Rejected

Lasso

Table 11:T-test Lasso Regression H0 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx - Clean dataset	t = -15.04
	p = 2.09e-49
		Rejected

0.15_f_train_noisy.xlsx - Clean dataset	t = 7.22
	p = 1.20e-12
		Rejected

0.35_c_train_noisy.xlsx - Clean dataset	t = -39.56
	p = 5.57e-286
		Rejected

035_f_train_noisy.xlsx - Clean dataset	t = 14.67
	p = 3.58e-47
		Rejected

0.50_c_train_noisy.xlsx - Clean dataset	t = -61.23
	p = 0.0
		Rejected

0.50_f_train_noisy.xlsx - Clean dataset	t = 18.98
	p = 1.65e-7
		Rejected

Ridge

Table 12: T-test Ridge Regression H0 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx - Clean dataset	t = -15.03
	p = 2.42e-49
		Rejected

0.15_f_train_noisy.xlsx - Clean dataset	t = 7.22
	p = 1.18e-12
		Rejected

0.35_c_train_noisy.xlsx - Clean dataset	t = -39.54
	p = 8.26e-286
		Rejected

035_f_train_noisy.xlsx - Clean dataset	t = 14.68
	p = 3.25e-47
		Rejected

0.50_c_train_noisy.xlsx - Clean dataset	t = -61.20
	p = 0.0
		Rejected

0.50_f_train_noisy.xlsx - Clean dataset	t = 18.98
	p = 1.55e-76
		Rejected

As it is shown from the tables, Null hypothesis H0 is rejected for the datasets *Air-Quality*, means that all regression models don't have the same outcomes when they were trained with clean and when they were trained with noisy data.

6.3.1.2. Computer Hardware

At the next tables are displayed the results from t-test for the *computer-hardware* dataset. Same results as the ones found in previous chapter.

Linear

Table 13: T-test Linear Regression H0 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - Clean dataset	t = -2.59 p = 0.02	Rejected
0.15_f_train_noisy.xlsx - Clean dataset	t = 1.71 p = 0.17	Accepted
0.35_c_train_noisy.xlsx - Clean dataset	t = -7.78 p = 3.77e-11	Rejected
035_f_train_noisy.xlsx - Clean dataset	t = 3.64 p = 0.0009	Rejected
0.50_c_train_noisy.xlsx - Clean dataset	t = -9.94 p = 1.88e-15	Rejected
0.50_f_train_noisy.xlsx - Clean dataset	t = 5.26 p = 2.22e-06	Rejected

Lasso

Table 14: T-test Lasso Regression H0 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx
-
Clean dataset
	t = -2.60
	p = 0.02
		Rejected

0.15_f_train_noisy.xlsx
-
Clean dataset
	t = 1.71
	p = 0.179
		Accepted

0.35_c_train_noisy.xlsx
-
Clean dataset
	t = -7.78
	p = 3.75e-11
		Rejected

035_f_train_noisy.xlsx
-
Clean dataset
	t = 3.64
	p = 0.0009
		Rejected

0.50_c_train_noisy.xlsx
-
Clean dataset
	t = -9.95
	p = 1.84e-15
		Rejected

0.50_f_train_noisy.xlsx
-
Clean dataset
	t = 5.26
	p = 2.24e-06
		Rejected

Ridge

Table 15: T-test Ridge Regression H0 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx
-
Clean dataset
	t = -2.59
	p = 0.02
		Rejected

0.15_f_train_noisy.xlsx
-
Clean dataset
	t = 1.71
	p = 0.17
		Accepted

0.35_c_train_noisy.xlsx
-
Clean dataset
	t = -7.78
	p = 3.77e-11
		Rejected

035_f_train_noisy.xlsx
-
Clean dataset
	t = 3.64
	p = 0.0009
		Rejected

0.50_c_train_noisy.xlsx
-
Clean dataset
	t = -9.94
	p = 1.88e-15
		Rejected

0.50_f_train_noisy.xlsx
-
Clean dataset
	t = 5.26
	p = 2.22e-06
		Rejected

6.3.1.3. Facebook Metrics

The pattern continuous also into this dataset.

Linear

Table 16: T-test Linear Regression H0 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx
-
Clean dataset
	t = -11.95
	p = 7.68e-25
		Rejected

0.15_f_train_noisy.xlsx
-
Clean dataset
	t = 4.90
	p = 3.96e-06
		Rejected

0.35_c_train_noisy.xlsx
-
Clean dataset
	t = -18.42
	p = 1.68e-44
		Rejected

035_f_train_noisy.xlsx
-
Clean dataset
	t = 6.99
	p = 8.02e-11
		Rejected

0.50_c_train_noisy.xlsx
-
Clean dataset
	t = -23.58
	p = 4.11e-59
		Rejected

0.50_f_train_noisy.xlsx
-
Clean dataset
	t = 9.63
	p = 5.79e-18
		Rejected

Lasso

Table 17: T-test Lasso Regression H0 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - Clean dataset	t = -12.01 p = 5.06e-25	Rejected
0.15_f_train_noisy.xlsx - Clean dataset	t = 4.90 p = 3.96e-06	Rejected
0.35_c_train_noisy.xlsx - Clean dataset	t = -19.96 p = 5.11e-49	Rejected
035_f_train_noisy.xlsx - Clean dataset	t = 7.03 p = 6.38e-11	Rejected
0.50_c_train_noisy.xlsx - Clean dataset	t = -24.55 p = 1.08e-61	Rejected
0.50_f_train_noisy.xlsx - Clean dataset	t = 9.70 p = 3.68e-18	Rejected

Ridge

Table 18: T-test Ridge Regression H0 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - Clean dataset	t = -11.95 p = 7.68e-25	Rejected
0.15_f_train_noisy.xlsx - Clean dataset	t = 4.88 p = 4.19e-06	Rejected
0.35_c_train_noisy.xlsx - Clean dataset	t = -18.42 p = 1.68e-44	Rejected
035_f_train_noisy.xlsx - Clean dataset	t = 6.99 p = 8.02e-11	Rejected
0.50_c_train_noisy.xlsx - Clean dataset	t = -23.58 p = 4.11e-59	Rejected
0.50_f_train_noisy.xlsx - Clean dataset	t = 9.63 p = 5.79e-18	Rejected

In general, apart from one cases (apart from one case (15% noise on the features), null hypothesis H0 is rejected for all datasets, thus Given the characteristics of each learning technique, noisy training data, does affect the final prediction of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 datasets and the noisy versions of them.

6.3.2. Hypothesis testing for Null hypothesis – H00

We run H00[2] hypothesis testing between the predicted values from a model which has been trained feature noisy data and the predicted values from a model which has been trained with target noisy data. We run tests for each noise level:15%, 35%, 50%, 3 regression models: Linear Regression, Lasso Regression, Ridge Regression and 3 datasets.

6.3.2.1. Air-quality

Linear

Table 19: T-test Linear Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 15.03
	p = 2.42e-49
		Rejected

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = 39.54
	p = 8.26e-286
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 61.20
	p = 0.0
		Rejected

[2] *Null hypothesis – H00:* Given the characteristics of each learning technique, we initially propose that feature noise and target noise in training data, affect the same the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 2 different noise levels of each dataset.

Lasso

Table 20: T-test Lasso Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = -15.04
	p = 2.09e-49
		Rejected

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = -39.56
	p = 5.57e-286
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = -61.23
	p = 0.0
		Rejected

Ridge

Table 21: T-test Ridge Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = -15.03
	p = 2.42e-49
		Rejected

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = -39.54
	p = 8.26e-286
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = -61.20
	p = 0.0
		Rejected

6.3.2.2. Computer Hardware

Linear

Table 22: T-test Linear Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 1.71
	p = 0.17
		Accepted

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t =3.64
	p = 0.0009
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 5.26
	p = 2.22e-06
		Rejected

Lasso

Table 23: T-test Lasso Regression H00 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 1.71
	p = 0.17
		Accepted

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = 3.645
	p = 0.0009
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 5.26
	p = 2.24e-06
		Rejected

Ridge

Table 24: T-test Ridge Regression H00 null hypothesis

Datasets	t-test	Null hypothesis

0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 1.71
	p = 0.17
		Accepted

0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = 3.64
	p = 0.0009
		Rejected

0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 5.26
	p = 2.22e-06
		Rejected

Except the case of adding 15% noise on the features, Null hypothesis H00 is also rejected for the datasets *computer-hardware*. All regression models have different outcomes for different noise type on the same noise level. At 15% feature noise, we have already seen that the outcomes of the regression models don't affect a lot, see Sect 6.3.1.2. At the case which H0 is accepted, the two predicted populations have identical mean.

6.3.2.3. Facebook Metrics

Linear

Table 25: T-test Linear Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = -4.88 p = 4.19e-06	Rejected
0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = -6.99 p = 8.02e-11	Rejected
0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = -9.63 p = 5.79e-18	Rejected

Lasso

Table 26: T-test Lasso Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 4.90 p = 3.96e-06	Rejected
0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = 7.03 p = 6.38e-11	Rejected
0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 9.70 p = 3.68e-18	Rejected

Ridge

Table 27: T-test Ridge Regression H00 null hypothesis

Datasets	**t-test**	**Null hypothesis**
0.15_c_train_noisy.xlsx - 0.15_f_train_noisy.xlsx	t = 4.88 p = 4.19e-06	Rejected
0.35_c_train_noisy.xlsx - 0.35_f_train_noisy.xlsx	t = 6.99 p = 8.02e-11	Rejected
0. 5_c_train_noisy.xlsx - 0. 5_f_train_noisy.xlsx	t = 9.63 p = 5.79e-18	Rejected

In general, apart from one cases (apart from one case (15% noise on the features), null hypothesis H00 is rejected for all datasets, given the characteristics of each learning technique, the feature noise and target noise in training data, don't affect equally the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 2 different noise levels of each dataset.

6.3.3. Hypothesis testing for Null hypothesis – H000

We run H000[3] hypothesis testing between the predicted values from a model which has been trained with data that has 35% noise and a model which has been trained with data include 50% noise. We run tests for two noise types: target noise, feature noise, 3 regression models: Linear Regression, Lasso Regression, Ridge Regression and 3 datasets.

6.3.3.1. Air-quality

Linear

Table 28: T-test Linear Regression H000 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = -24.85
	p = 4.88e-126
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = 4.14
	p = 6.92e-05
		Rejected

Lasso

Table 29: T-test Lasso Regression H000 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = 24.87
	p = 2.97e-126
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = -4.14
	p = 6.83e-05
		Rejected

[3] *Null hypothesis – H000:* Given the characteristics of each learning technique, we initially propose that the high level of noise in training data, doesn't affect negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 clean datasets and the noisy versions of them.

Ridge

Table 30: T-test Ridge Regression H000 null hypothesis

Datasets	t-test	Null hypothesis

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = 24.85
	p = 4.88e-126
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = -4.14
	p = 6.92e-05
		Rejected

6.3.3.2. Computer Hardware

Linear

Table 31: T-test Linnear Regression H000 null hypothesis

Datasets	t-test	Null hypothesis

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = 3.47
	p = 0.001
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = -2.28
	p = 0.05
		Rejected

Lasso

Table 32: T-test Lasso Regression H000 null hypothesis

Datasets	t-test	Null hypothesis

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = -3.47
	p = 0.001
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = 2.28
	p = 0.05
		Rejected

Ridge

Table 33: T-test Ridge Regression H000 null hypothesis

Datasets	t-test	Null hypothesis
0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = -3.47 p = 0.001	Rejected
0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = 2.28 p = 0.05	Rejected

6.3.3.3. Facebook Metrics

Linear

Table 34: T-test Linear Regression H000 null hypothesis

Datasets	t-test	Null hypothesis
0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = -6.51 p = 1.19e-09	Rejected
0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = 4.71 p = 9.03e-06	Rejected

Lasso

Table 35: T-test Lasso Regression H000 null hypothesis

Datasets	t-test	Null hypothesis
0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = 7.18 p = 2.72e-11	Rejected
0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = -4.71 p = 9.03e-06	Rejected

Ridge

Table 36: T-test Ridge Regression H000 null hypothesis

Datasets	**t-test**	**Null hypothesis**

0.35_c_train_noisy.xlsx - 0.50_c_train_noisy.xlsx	t = 6.51
	p = 1.19e-09
		Rejected

0.35_f_train_noisy.xlsx - 0.50_f_train_noisy.xlsx	t = -4.71
	p = 9.03e-06
		Rejected

In general, null hypothesis H000 is rejected for all datasets, given the characteristics of each learning technique, the high level of noise in training data, affects negatively the predictions of regression algorithms: Linear Regression, Ridge Regression and Lasso Regression, using 3 datasets and the noisy versions of them.

7. Source Code

For the implementation of the regression models and the execution of statistical tests we have implemented some python scripts. Looking on the project there are 3 subfolders, one per dataset, inside of these subfolders there are the generated noisy datasets at the path /noisy_datasets and 2 python scripts at which we run the t-test.

For example, for the *Air-Quality dataset*:

- Air-H0.py →train regression models with clean and noisy dataset and calculate t-test for hypothesis H0
- Air-H00.py → train regression models with various levels of noisy data or different types of noisy data and calculate t-test for hypothesis H0 and H00. Running the script, you should define from command lien which hypothesis testing you would like to run

8. Conclusions

Running three statistical tests we conclude that noisy data affect the final predictions of a regression model. Comparing the outcomes from the original datasets with the outcomes from the noisy dataset using the two-sample paired t-test and 3 datasets from different domains, we conclude that adding noise either on target feature or on the remaining features of the dataset, the noise in the end will affect the outcomes of the regression problem. More specifically, the RMSE is getting bigger and bigger and the “fit” of the regression line is getting worst. The higher the uncertainty and the appearance of the noise in the dataset, the less accurate the model will be. Another important notice is that noise on target feature altered the values far more compare to the noise that was inducted into features, increasing the RMSE by a wider margin.

8.1. Repository

Code is available on GitHub on that link: https://github.com/JoHNNyB92/applied/

References

1. A study of the effect of different types of noise on the precision of supervised learning techniques - David F. Nettleton · Albert Orriols-Puig · Albert Fornells, https://link.springer.com/article/10.1007/s10462-010-9156-z

2. Noisy and Missing Data Regression: Distribution-Oblivious Support Recovery - Yudong Chen, Constantine Caramanis, http://proceedings.mlr.press/v28/chen13d.pdf

3. The Truth About Linear Regression 36-350, Data Mining 21 October 2009, https://www.stat.cmu.edu/~cshalizi/350/lectures/17/lecture-17.pdf

4. Scattered Data Approximation of Noisy Data via Iterated Moving Least Squares - Gregory E. Fasshauer and Jack G. Zhang, https://pdfs.semanticscholar.org/9b7d/891601e006b85a3f49ef432f35524aa2a328.pdf

5. Machine Learning Algorithms, a study of noise sensitivity – Elias Kalapanidas, Nikolaos Avouris, Marian Craciun, Daniel Niagu, http://delab.csd.auth.gr/bci1/Balkan/356kalapanidas.pdf

6. A robust hybrid of lasso and ridge regression Art B. Owen Stanford University October 2006, http://statweb.stanford.edu/~owen/reports/hhu.pdf

7. Online Regression with Controlled Label Noise Rate, Edward Moroshko and Koby Crammer Department of Electrical Engineering The Technion, Haifa, Israel, http://ecmlpkdd2017.ijs.si/papers/paperID459.pdf

8. A comparative study of linear regression methods in noisy environments, Marco S.Reis* and Pedro M.Saraiva, https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.897

9. A significance test for Lasso, Richard Lockhart,[2] Jonathan Taylor,[3] Ryan J. Tibshirani,[4] and Robert Tibshirani[5,] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4285373/

10. http://archive.ics.uci.edu/ml/datasets/air+quality

11. https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

12. https://archive.ics.uci.edu/ml/datasets/Facebook+metrics

13. https://courses.washington.edu/b515/l5.pdf

14. https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/10/lecture-10.pdf

15. https://stattrek.com/regression/slope-test.aspx

16. http://www.econ.nyu.edu/user/ramseyj/textbook/chapter11.pdf

17. https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/t-test/#PairedTTest

18. https://towardsdatascience.com/inferential-statistics-series-t-test-using-numpy-2718f8f9bf2f