Analysis of Interpretability and Explainability of Regression approaches

Sushma Anand Akoju
sushma.akoju@colorado.edu
University of Colorado Boulder
Boulder, USA

Prof. Brian Zaharatos
BrianZaharatos
University of Colorado Boulder
Boulder, USA

KEYWORDS

Interpretability, Explainability, Reasoning

1 PROBLEM, GOAL AND BACKGROUND

1.1 Problem

As part of the Linear regression chapters and course content covered in STAT 5010 course, it seems that the regression analysis has been well known for explaining numerical data analysis and the mathematical interpretations, explanations. Lasso and Ridge Regression, Support Vector Regression are Machine Learning algorithms used with optimization and hyper parameter tuning to address problems from data that are not solved by Linear Regression models.

1.2 Goal

The goal is

(1) To find how interpretable, explainable a Linear regression analysis is when compared to Machine Learning models, Neural Networks.
(2) To define more specifically, the 3.1 Approach section provides details of how this is organized as per feedback and discussion and reflections from data analysis, discussions from Prof. Brian Zaharatos combined with STAT 5010 course lectures.

2 MOTIVATION AND BACKGROUND

Machine Learning has always been integral part of Artificial Intelligence. Machine Learning draws on concepts and results from many fields including statistics, artificial intelligence, philosophy, information theory, biology, cognitive science, computational complexity and control theory. Machine Learning is the study of algorithms that allow computer programs automatically improve through experience. [6] [14] Ever since Artificial Intelligence (AI) as a thought experiment was started, it has been common philosophical quest to find out the ethical, legal and societal concerns of Artificial Intelligence applications. [17]. Artificial Intelligence applications are vastly used in several real world applications more recently. However there seems to be some examples that serve as the unexpected consequences in the real world AI applications. The machine bias risk assessment in criminal sentencing https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing https://github.com/propublica/compas-analysis and racial bias in Machine learning algorithms [12] have resulted in research and analysis of the underlying problem of fairness and bias in Machine Learning algorithms and the datasets that served as input to Machine Learning algorithms. One such consequence also led to landmark sentencing reform bill pending in Congress, that would mandate use of combined assessments of future risk for a repeated crime during the process of sentencing at federal prisons. While the bills themselves take time, there are other serious consequences. The most dire consequence seems to be loss of trust in the tools used by the criminal justice system and the system itself, that was built over several years of various well founded judicial reforms. Another important consequence is the loss of trust in AI systems assumed to be built on the basis of such theoretical, analytical foundations.

The modern applications of Artificial Intelligence (AI) are widely used in the domains such as marketing, psychology, social sciences, healthcare and law. Each one of the domains have significant number of users or a group of users such as a research community or an organization who sometimes consume resulting analytical summary of an AI application and/or generate such data which serves as input to an AI system or have more complex, indirect participation in such data-driven AI systems. The most common examples are the credit reporting, patient diagnosis reports. Thus such participatory nature of the complex AI tasks, would require to communicate the intermediate steps and results to users for interpretation, analysis i.e. explanatory purposes.
With such aforementioned motivation, I attempt to learn, analyze, define the explainability and interpretability for each of the algorithms i.e. starting from Linear regression as a statistical analytical tool for data analysis to more recent Machine Learning algorithmic analysis as well as Deep learning approaches that are well founded in mathematical and statistical analysis.

2.1 Why are you interested in this problem?

From the aforementioned motivation and background, I think the problem of interpretability and explainability seems to exist in data analysis. [2]. With a growing popularity and success from Deep Learning models in building vast number of applications, from several domains and for various types of downstream and upstream tasks, understanding deep learning models and the underlying causal analysis of outcomes from data analysis requires attention. It is not necessarily a mandatory exercise to spend time to understand a Deep learning model's interpretability, explainability to provide explanations for analysis from training of a model. More recently with the advent of Transfer learning, the analysis of the data seems to leave much to be desired to understand a deep learning model in understanding data. [21] For a given domain D_s with a corresponding source task T_s and target domain D_t with a corresponding target task T_t, transfer learning is a process of improving the target predictive function $f_t(.)$ by using the related information i.e weights learned from learning task T_s over domain D_s, where $D_s \neq D_t$ and $T_s \neq T_t$. [21]. Thus transfer learning, which is vastly used, can provide lot of background information, context and analysis of any underlying issues in model outcomes and analyses. So due to this, first I would like to analyze and understand why the interpretability and explainability is well defined in statistical model, how interpretability and explainability expands over Machine Learning models and finally analyze how the interpretability and explainability can be defined over Deep Learning models.

2.2 What is the relevant background information for readers to understand your project? Assume that your audience is not an expert in the application

The motivation and background as discussed in Motivation and Background section provides examples of analysis for biases in data and analysis. For example investigating gender bias in a popular Natural Language processing Deep Learning Model known as BERT (Bidirectional Encoder Representations from Transformers)[8] suggests intrinsic gender bias in Contextual Language Models [4]. BERT is a model that uses masks to replace words randomly and learn to predict the missing words from the contextual information. In an attempt to achieve that, predictions of NLP tasks can vary vastly by varying gender words such as replacing "he" with "she". In the gender bias case, suppose the sentence BERT is learning about is "$< Name >$ is angry". If we observe that BERT assigns higher or lower scores to this masked sentence when $< Name >$ is replaced with female names than that of male names, then this is called as gender bias in BERT model. [4]. It is possible to have such instance from textual data such as when the corpus has a context from a fictional novel, where the use of word anger is associated with more female names than male names which could be context-sensitive information. There is a vast research in addressing implicit as well as explicit bias examples in Deep Learning Language models. As the textual corpus increases and as the Language models are trained over larger datasets, it becomes more difficult to address such biases w.r.t race, gender and any other biases. Such a bias can be serious in nature if the corpus is a patient's Electronic Health record and that could impact a resulting diagnoses' quality. Thus it becomes a important to assess and find explainability, interpretability in general for deep learning models.

Deep learning models are known to be black box in nature and vast amount of research is actively pursued in this direction. [20]

2.3 Is there any prior research on your topic that might be helpful for the audience?

There is a lot of research actively going on more towards modern Artificial Intelligence solutions that do use Deep Learning algorithms. More recently in conferences, data analysis communities, there are separate workshops and tutorials on Interpretability and Explainability on Black-box models that are vastly used in various downstream and upstream tasks in Data Analysis for vast amoutns of data. https://aaai.org/Conferences/AAAI-22/ws22workshops/ws18

2.4 From where did the data come? Is this an experiment or observational study? Who collected the data? Why was the data collected (if you weren't the one doing the collecting)

This dataset was collected by Experimental Analysis and Modeling of Frictional Behavior of Lavender Flowers (Lavandula stoechas L.) and is available at https://users.stat.ufl.edu/ winner/data/lavender_friction_SFC.txt. The period from June 2012 to September 2012. The dataset includes 11 attributes and 1 output attribute (class). [19] The variables details from the dataset are as follows:

(1) surface
(2) moisture
(3) Static friction coefficients (DFC)

I chose this dataset because, I want to demonstrate the interpretable and explainable nature of Linear regression over a simple linear regression dataset and find insights from the analysis. Finally I want to discuss, analyze the interpretable and explainable nature of other approaches, algorithms for comparison.

First I would be explaining a dataset from simulated dataset. For navigating between regression algorithms and Neural networks, I want to use XOR table simulation as input data for discussing Universal approximation theorem and how regression, Neural networks and Deep Learning models work over non-linear separable data.

2.5 What are the questions of interest that you hope to answer?

(1) Can we infer Linear regression analysis easier to interpret and explain the causality? The assumption is Linear regression is easier to interpret, explain than other regression approaches. I looked at one of publications that I hope to summarize my analysis and learning from this reference https://www.ripublication.com/ijaer17/ijaerv12n20_77.pdf
(2) How interpretable and explainable are Machine Learning algorithms?
(3) Are Neural Networks interpretable and explainable?
(4) How interpretable and explainable are deep learning models in comparison to other approaches?
(5) Why does each of above approaches have different results when all of them have strong mathematical and statistical analysis? What makes each of above algorithms, models different from Regression Analysis?
(6) Can Neural Networks and Deep Learning models find intersection in terms of interpretability, explainability with respect to Linear regression?

3 APPROACH, DEFINITIONS

3.1 Approach

In this project, I attempt to analyze, formulate how interpretability and explainability evolved starting from Linear Regression, advanced Regression algorithms to until Deep Learning networks. First, I begin by defining interpretability and explainability in Artificial Intelligence as two different terms [9], then extend this more specifically towards Statistical definitions, analysis. I further attempt to concretize the idea in the context of Machine Learning, Neural Networks, discussing the reasons, importance of the interpretability and explainability in data analysis. [15] [18] [9] [5] [22]. The approach involves following steps:

1. To first define generalized specifications of interpretability and explainability.
2. To write each of the algorithm's respective notation, approach, metrics for measure of performance, analysis to introduce the concept and context.
3. To summarize and discuss benefits and drawbacks of each of the algorithms. Then define interpretability and explainability in terms of statistical and machine learning terminology.
4. To discuss each of the algorithms with data analysis and the results in terms of interpretability and explainability.

4 PROBLEM DEFINITION

In this project, I attempt to analyze, formulate how interpretability and explainability evolved starting from Linear Regression, advanced Regression algorithms to until Deep Learning networks. First, I begin by defining interpretability and explainability in Artificial Intelligence as two different terms [9], then extend this more specifically towards Statistical definitions, analysis. I further attempt to concretize the idea in the context of Machine Learning, Neural Networks, discussing the reasons, importance of the interpretability and explainability in data analysis. [15] [18] [9] [5] [22].

4.1 Introduction

4.2 Approach

The approach involves following steps:

1. To first define generalized specifications of for each approach.
2. To define metrics used for comparing each of the approaches.
3. To write each of the algorithm's respective notation, approach analysis to introduce the concept and context.
4. To summarize and discuss benefits and drawbacks of each of the approach. Then define interpretability and explainability in terms of statistical and machine learning terminology.
5. To discuss each of the algorithms with data analysis and the results in terms of interpretability and explainability.

4.3 Dataset

I selected dataset [19] for this study. For navigating between regression algorithms and Neural networks, I attempt to use XOR table simulation as input data for discussing Universal approximation theorem.

4.3.1 *About the SFC Dataset.*

Lavender flowers are well known for the medicinal properties through history. They are indigenous to mountain areas from western European part of Mediterranean region. The SFC dataset was selected from https://users.stat.ufl.edu/~winner/datasets.html. Reason for selecting this dataset, is to have true Linear Regression dataset that has been proven to have Linear regression. [19]

Coefficient of friction of plant on the various surfaces is needed in designing of silos, storage of agricultural products and handling equipment, such as conveyors, and design of other equipment used in post harvest processing. [11] Frictional properties of agricultural products are important parameters for designing handling machines such as packaging, cleaning, conveyors, augers and other equipment. [19] Thus Lavender flowers are considered medicinal plants and several products are manufactured. The dataset was originally collected to study the lavender flowers and the surface.

1. Static Frictional Coefficient (SFC) is the response variable.
2. Moisture and the surface are the predictors.
3. Frictional force is $F = \mu N$, where F is Frictional force, N is normal force and μ is SFC or DFC. In this case it is SFC.
4. Defining the Static Fricton force: The static friction force is a force which exists between two static objects.
5. The goal of the paper published was to precisely measure SFC as affected by moisture content and contact surface. $SFC = a(MC) + b$
6. Lavender flowers have following physical properties:
   (a) Length mm 24.6 ± 3.774
   (b) Width mm 9.9 ± 1.921
   (c) Thickness mm 5.7 ± 0.781
   (d) GMD mm 11.053 ± 1.045
   (e) Sphericity
   (f) Mass g 0.265 ± 0.013

4.3.2 *Discussion from Research work [19].*

The authors in the paper are attempting to analyze and find out how friction coefficient can be influenced from moisture content and contact surface for medicinal herbs and specifically for Lavender flowers (Lavandula stoechas L.).

Coefficient of friction of plant on the various surfaces is needed in designing of silos, storage of agricultural products and handling equipment, such as conveyors, and design of other equipment used in post harvest processing. [11] Frictional properties of agricultural products are important parameters for designing handling machines such as packaging, cleaning, conveyors, augers and other equipment. [19] Thus Lavender flowers are considered medicinal plants and several products are manufactured. The dataset was originally collected to study the lavender flowers and the surface.

5 DEFINITIONS

5.1 Explainability

To answer the question of "why" serves as a simplest form of definition of explanation. Philosophers have explored and debated what constitutes as good explanation. The two why-questions of interest are why and why should. The questions of interests in terms of explainable planning literature extends to why shouldn't and why should. Additionally inference could serve as a best explanation as many say and similar views are extended towards abductive reasoning. [9] Definition of abductive reasoning itself suggests inference to the best explanation or the most likely explanation as a conclusion over a set of observations.

More recently, explainability has taken multiple directions:

(1) Explaining something statistically and mathematically that supports hypothesis over Deep learning model activations, epochs and layers.
(2) Explaining something visually to a data scientist or someone who has no idea about Deep learning as well as the Language processing tasks.
(3) Trying to explain everything is going to be challenging anyway, just limit explainability to intersect between mathematical as well as visual explanations.
(4) Use other more advanced approaches such as higher order logic (is research intensive) and is one direction I am in favor of. An example of this direction is [7] where the authors propose that a network can learn to explain as it learns to classify by making use of First Order Logic.

More specifically [16] suggest explanation methods into 3 categories:

(1) Rule-extraction methods (deals with extracting rules that approximate the decision making process)
(2) Attribution methods (deals with measuring importance of a component by changing the input or internal components)
(3) Intrinsic methods (deals with interpretability, to understand and interpret internal structure of representations)

Rule extraction as per [16] can broadly be acheived by 3 subcategories:

(1) Decompositional approach (deals with breaking down the network into individual parts)
(2) Pedagogical approach (deals with viewing rule extraction itself as a learning task where target concept is the function computed by the network and input features are the network's input)
(3) Eclectic approach (deals with "..membership in this category is assigned to techniques which utilize knowledge about the internal architecture and/or weight vectors to complement a symbolic learning algorithm")

5.2 Interpretability

Explanation can be evaluated in two ways: according to its interpretability and completeness. interpretability is the degree to which a human can understand the cause of a decision, as a simplest form of definition. The goal of interpretability is to describe the features of a system in a way that is understandable by the humans. [9]

The goal of completeness, on the other hand, is to describe the operation of a system in an accurate way. Such context of interpretability and completeness can suggest explanation is more complete when it allows behaviour of system to be anticipated in more situations. [9]

A simplest example for this would be how the approach of defining each of algorithm in this project itself in terms of mathematical notation, steps, parameters, metrics serves as an explanation of the algorithm. The question then is how does the algorithm become interpretable, complete and hence explainable and provide explanations of results.

5.3 Universal Approximation Theorem

6 REGRESSION APPROACHES AND METRICS

6.1 Metrics

For this project analysis, following metrics will be used for comparison between each of the Regression approach:

(1) Root Mean Squared Error metrics suggest how far predicts fall from measured true values using Euclidean Distance. $\sqrt{\frac{\sum_{i=1}^{n}||(y_i-\hat{y}_i)||^2}{N}}$
(2) AIC and BIC: AIC is given as

$$AIC\left(g\left(\mathbf{x};\widehat{\boldsymbol{\beta}}\right)\right) = 2(p+1) + n\log(RSS/n).$$

BIC, is given as

$$BIC\left(g\left(\mathbf{x};\widehat{\boldsymbol{\beta}}\right)\right) = (p+1)\log(n) - 2\log L\left(\widehat{\boldsymbol{\beta}}\right),$$

(3) RSquared = $1 - \frac{RSS}{TSS}$, where RSS is Residual Sum of Squares and TSS is Total Sum of Squares.
(4) Mean Squared Prediction Error
(5) Variance Inflation factor $Var(\hat{\beta}) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_{ij}-x_j)^2)}\frac{1}{1-R_j^2}$ and $VIF = \frac{1}{1-R_j^2}$ where R_j^2 is Coefficient of determination. If $VIF \geq 5$ then there is some evidence of multicollinearity. If $VIF \geq 10$ then there is strong evidence of multicollinearity.
(6) Condition number of (X^TX) is $\mathbb{K} = \sqrt{\frac{\lambda_{largest}}{\lambda_{smallest}}}$

6.2 Linear Regression

(1) Theory and Notation: Given a vector of inputs $X = (X_1, X_2, ..., X_p)$, predict a real valued output Y. A general form of Linear model is: $Y = f(X_1, X_2..., X_p) + \varepsilon$. For a 2 predictor vector inputs, $X = (X_1, X_2)$ then simple linear regression model can be written as $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon_i$.
Error is the difference between prediction and outcome. Errors are assumed to have Gaussian Distribution. Coefficients are the learned feature weights or coefficients of the predictors.

$$\begin{pmatrix} y_0 \\ y_1 \\ ... \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ 1 & .. & .. \\ 1 & x_{1n} & x_{2n} \end{pmatrix} + \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ ... \\ \varepsilon_n \end{pmatrix} \tag{1}$$

The assumptions of Linear Regression model are as follows:
(a) Linearity $Y = \beta X$
(b) Independence $cov(Y_i, Y_j) = 0, i \neq j$
(c) Homoskedasticity (Constant Variance) $var(Y_i) = \sigma^2$
(d) Normality $Y_i \sim N(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + .. + \beta_p X_{ip}, \sigma^2)$

6.2.1 Gauss-Markov Theorem. The assumptions are:

- $E[\epsilon] = 0$
- $V[\epsilon] = \sigma^2$
- $Cov[\epsilon_i, \epsilon_j] = 0 \; \forall \; i \neq j$

6.2.2 Benefits and Drawbacks. From the analysis over [19] and data analysis from https://tinyurl.com/5n6es9tx the advantages of the Linear Regression:

(1) Approach and theory are simple and easy to understand.
(2) If a model has overfitting or underfitting or has performed poorly, it is easier to understand and interpret.
(3) Assumptions of linearity make it easier to see analyze if model follows a linear regression or not.
(4) More generally, so far the linearity is not violated, linear models provide reasonable, truthful explanations.
(5) From the dataset, it was easier to determine if features have multicollinearity or not.
(6) Relationship between the features, correlation and causal dependence between predictors and response variables are simpler to understand.
(7) If Gauss Markov Theorem assumptions are satisfied, Ordinary Least Squares is best linear unbiased estimator for the dataset.

The disadvantages of Linear regression are as follows:

(1) Assumptions from Linear regression seems to be strict.
(2) Feature selection, feature engineering seems to be difficult although hypothesis and analysis can be well explained.
(3) Linear models are sensitive to outliers.
(4) Advantages of linear regression are limited to only Linearity assumptions.
(5) Gauss Markov assumptions hold true if dataset is small. The assumptions fail to work over larger datasets, generally.
(6) (X^TX) must be invertible and condition must be true about the columns of X for the "Gram" matrix X^TX to be invertible.
    (a) The columns must be linearly indedependent for the gram matrix X^TX to be invertible. i.e if columns in X are linearly independent, there exists a vector u, such that $X^TXu = 0$. Thus X is a full column rank matrix. counter example: one column is a constant multiple of another column, where gram matrix becomes non-invertible.
    (b) The features i.e. X_1, X_2... are independent and there is zero or very less correlation between features. Change in one feature does not lead to change in another feature. This also means, one feature cannot be expressed in terms of another feature. Maybe in most regression analysis, the correlation between features is expected to be less than 10
    (c) X^TX is a (p+1) x (p+1) matrix , where p is number of predictors. If n is less than number of model parameters (p+1), then X^TX may become a singular matrix and non-invertible, since there will not be a unique solution. At practical level, this misleads to project linear dependencies among columns i.e. predictors. To explain this , I looked at the Gram matrix properties and conditions of Gram determinant. Predictors can be thought of as vectors in R^n, then square of n-dimensional volume of n-dimensional parallelotope formed by these vectors is the Gram determinant. Predictors will be linrealy independent if and only if n-dimensional parallelotope has non-zero volumne, Gram determinant must be non-zero, Gram matrix is non-singular. Thus if n = p+1, then gram determinant reduces to n-dimensional volume. volume of n-dimensional parellelotope should not be larger than volumes of complementary faces. Thus the volume of parallelotope could become zero if n « p+1. In practical cases, if less measurementts exist than predictors such that number of predictors are not too large, it seems to become difficult to establish linear independence but not the presence of linear dependence itself.
    (d) with Gram matrix$\hat{\beta} = (X^TX)^{-1}X^TY$, if X^TX is not invertible, then there will not be unique solution for $\hat{\beta}$ because $(X^TX)^{-1}$ does not exist.

6.3 Support Vector Regression

(1) Theory and Notation: SVM has the constraint

$$\sum_i y_i \alpha_i = 0, \tag{2}$$

which makes the total weight for the positive class equal to that of the negative class. The Parzen classifier automatically satisfies this constraint since the total weight for each class is one.

7 SEPARABLE CASE

We are given data

$$(x_i, y_i), \quad x \in R^d, y \in \{-1, 1\}.$$

We want a linear classifier in an infinite-dimensional kernel space,

$$g(x) = \operatorname{sign}(\phi(w) \cdot \phi(x) + b), \tag{3}$$

where

$$\phi(w) \cdot \phi(x) = K(w, x). \tag{4}$$

Support Vector Regression (SVR) is characterized by kernels, control over margins (Vapnik-Chervonenkis) and number of support vectors. SVR treats regression problem as a generalization of classification and thus it estimates a continuous valued Multivariate function. [3]. SVR adapts Karush-Kuhn-Tucker (KKT) conditions.

For the purpose of scope of this project as it is towards Linear regression, I have only included the Linear seperable case. SVM can be adapted to non-linear optimal decision surfaces for finding an optimal seperating hyperplane.

7.0.1 Benefits and Drawbacks. The benefits of Support Vector Regression are as follows:

(1) Support Vector regression is less popular but vastly used for real valued function estimation. SVR trains a symmetrical loss function which penalizes extreme cases of mistakes in regression i.e. higher and lower misestimates. [3]
(2) computational complexity does not depend on dimentionality of input space.
(3) SVR has reasonable generalizability with high prediction accuracy.

The benefits of Support Vector Regression are as follows:

(1) computational complexity does not depend on dimensionality of input space. This makes it difficult to explain the results.
(2) Training time is very high.
(3) it might become challenging if the outliers should not be considered within the "margin" since SVR considers outliers into regression and decision boundary.
(4) Multiclass classification requires multiple SVM models for classification.

7.1 Lasso and Ridge Regression

7.1.1 Regularization. Regularization is a method for solving ill-posed problems or problems of overfitting, as discussed in the Statistics course. This method involves introducing penalty term that provides additional information to the model. In case of LASSO and Ridge regression methods, the penalty imposes a shrinkage on the coefficient estimates of ordinary least squares. The penalty controls the instability found in least squares model with nonorthogonal matrices. L_p regularization term we have $L_p = (\sum_i \|\beta_i\|^p)^{\frac{1}{p}}$. Ridge and Lasso follow L_2 and L_1 regulatization. Regularization is used for best subset selection and step-wise subset selection.

$$(Y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta \tag{5}$$

$$= \hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y \tag{6}$$

At $\lambda = 0$, the ridge regression solution becomes Ordinary Least Squares solution. At $\lambda = \infty$, coefficients become zero.

Lasso is (expanded to) Least Absolute Shrinkage and Selection Operator. This vastly used for feature selection, regularization of feature weights.

$$(Y - X\beta)^T (Y - X\beta) + \lambda |\beta|_1 \tag{7}$$

7.1.2 Benefits and Drawbacks. The benefits of Lasso and Ridge regression are as follows: [10] [13]

(1) Lasso can be used when there is high multicollinearity, to automate variable elimination and for feature selection.
(2) In case of both Lasso and Ridge regression, data standardization is a common pre-requisite. [10]
(3) λ hyper parameter tuning is common technique to find optimal λ to optimal number of coefficients. Ridge regression coefficient solution is subject to size constraint so that largest possible positive coefficient does not cancel out a similarly large negative coefficient. [10]
(4) Singular Valued Decomposition (SVD) over centered values, helps with Ridge regression i.e. SVD of centered X is another way of expressing Principle Components of variables in X. More specifically degrees of freedom of λ evolved from Karush-Kuhn-Tucker conditions for the Ridge estimator. [10]
(5) If coefficients are approximately similar, ridge regression helps since it retains all coefficients in the model.
(6) If only small number of coefficients are approximately significant, Lasso regression helps since it can shrink insignificant coefficients.

The drawbacks of Lasso and Ridge regression are :

(1) It gets difficult to interpret coefficients as they are shrunk to zero.
(2) if coefficients are approximately similar, ridge regression helps since it retains all coefficients in the model.
(3) Lasso fails if pairwise correlations among predictors are too high since it tends to select only one coefficient among a group of highly correlated variables.
(4) Ridge regression trades variance for bias and vice versa in case of Lasso regression.

Artificial Neural Networks: Perceptron and Multi-layer Perceptron

A perceptron is a type of Artificial Neural Network. Perceptron is a unit that takes a vector of real valued inputs, calculates a linear combination of these inputs and then outputs a result of 1 if results is greater than a threshold, otherwise -1. [14]

Summary:
Learning a perceptron involves learning the weights w_0, w_1 ..., w_n . Thus the hypothesis space H consists of set of all possible candidates for weights i.e. real-valued weight vectors. [14]

$$H = \left\{ \overrightarrow{w} \middle| \overrightarrow{w} \in \mathbb{R}^{n+1} \right. \tag{8}$$

(1) Theory and Notation: More formally, a Linear Threshold Unit (LTU) given inputs x_1, through x_n, output $o(x_1, x_2, ...x_n)$ is computed as follows:

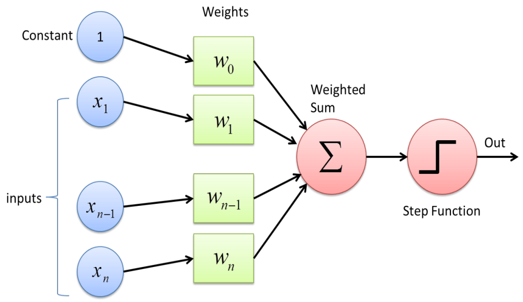
$$o(x_1, x_2, ...x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + w_2 x_2 + .. + w_n x_n > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where each w_i is a real-valued constant or a weight that determines the contribution of input x_i towards the output of the perceptron. w_0 is a threshold that weighted combinations of inputs $w_0 + w_1 x_1 + w_2 x_2 + .. + w_n x_n$ should be greater than so that the output of perceptron is 1.
This can be represented as a further simplified notation, as follows: [14]

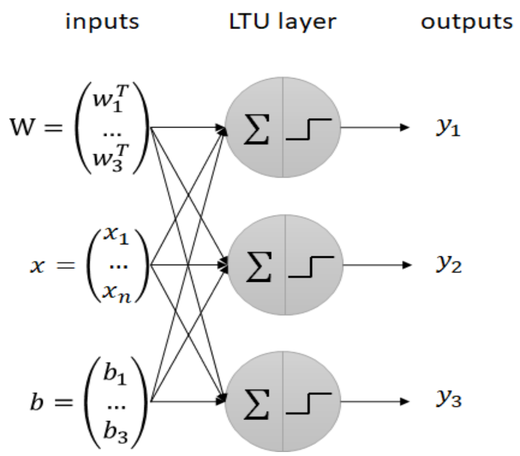
$o(\overrightarrow{x}) = sgn(\overrightarrow{w}.\overrightarrow{x})$

$$sgn(y) = \begin{cases} 1 & \text{if } y > 0 \\ -1 & \text{otherwise} \end{cases} \tag{10}$$

The following figure represents a Perceptron with Linear Threshold Unit.



The following figure example of 3 Linear Threshold Unit layers in a Perceptron.


inputs
LTU layer
outputs
$W = \begin{pmatrix} w_1^T \\ \dots \\ w_3^T \end{pmatrix}$
$x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$
$b = \begin{pmatrix} b_1 \\ \dots \\ b_3 \end{pmatrix}$
Σ
y_1
y_2
y_3

n is Number of samples and u is number of Linear Threshold Units. Input vector has a shape of $(n, 1)$. Weights vector has a shape of $(u, 1)$. Output vector has a shape of $(u, 1)$.
If we assume $f(x)$ be a linear function, then
$f(x) = sgn(w^T x - \Theta) = sgn(\sum_{i=1}^{n} w_i x_i - \Theta)$

7.1.3 About Perceptron. A single perceptron can represent many boolean functions. Let 1 be TRUE and let -1 be FALSE. For a 2 input perceptron, following weight values will give corresponding Boolean functions i.e Conjunction (AND) and disjunction (OR) operations using Perceptron. [14]

Operator	w_0	w_1	w_2	equation
AND	-0.8	0.5	0.5	$y = sgn(0.5x_1 + 0.5x_2 - 0.8)$
OR	-0.3	0.5	0.5	$y = sgn(0.5x_1 + 0.5x_2 - 0.3)$

Similarly, NAND $(\neg x_1 \wedge x_2)$ and NOR $(\neg x_1 \vee x_2)$ can also represented by perceptron. Thus to represent the NAND, for example, the above weights can be simply negated. Most linear functions are represented as a set of conjunctions or disjunctions or as an m-of-n representation using perceptron. **The m-of-n rules** is a generalization for a hypothesis space in which there are n literals x_1, x_2,...,x_n and a positive integer m, such that if m of the literals are true then m is true. This rule enables to represent the functions such as AND and OR very easily for setting m of n literals to equal weights (0.5) and setting a threshold.
Some Boolean functions are linearly separable. Simple conjunctions, disjunctions (some variables with negated representation of variables) can hold true for linear separability. Some of the examples of boolean functions/operations become more complex such as XOR, $(x_1 \wedge x_2) \vee (\neg x_1 \vee \neg x_2)$ which are linearly inseparable. Linearly inseparable functions cannot be represented using perceptron. [1]

(2) Algorithm: The pseudo code of the algorithm is as follows: Given x with p features, each feature consisting of n input samples. Let y be response vector with n observations. Let η be learning rate. Let o be output generated by perceptron.
Case 1 : Linearly Separable examples
   (a) Initialize weights $w_0 = w_1 = ... = w_n = 0$
   (b) For each input sample i from n samples,
      (i) Calculate output $o_i = sgn(w_t^T x_i$ The following is the perceptron training rule
      (ii) if $w_i(t+1) = w_i(t) - \eta(y_i - o_i)x_i$
      (iii) repeat until $\frac{1}{n}\sum_{i=1}^{n}|y_i - o_i|$ is less than a threshold γ
   (c) the algorithm converges if the data are linearly separable.
Case 2 : Linearly Non-Separable examples
   (a) Initialize weights $w_0 = w_1 = ... = w_n = 0$
   (b) For each input sample i from n samples,
      (i) Calculate output $o(x) = wx_i$ The following is the perceptron training rule
      (ii) if $w_i(t+1) = \eta(y_i - o_i)x_i$
      (iii) repeat until $\frac{1}{2}\sum_{i=1}^{n}(y_i - o_i)^2$ is minimized using Gradient Descent algorithm [14].

7.1.4 Benefits and Drawbacks. The drawbacks of Neural Networks are as follows:

(1) Multiclass classification which is difficult with Support Vector Machines can be easily implemented as it requires only single model.
(2) Neural networks encapsulate complex learning from various forms of data and can be adapted to different types of data.
(3) It has

The drawbacks of Neural Networks are as follows:

(1) Perceptron converges for linear separable examples, however fails to converge for examples that are not linearly separable.
(2) Neural networks are black box and it is often challenging to work through the process backwards from solution to all the until input layers.

8 APPROACH FOR DATA ANALYSIS

I first simulated data which satisfied assumptions of Linear regression. And I implemented a full analysis for Linear regression fit and diagnostics, which made it easier to adapt to the dataset and analysis. I then also conducted each of regression algorithm analysis over simulated data to understand the challenges.

I first conducted a data analysis by using visualization methods and found that Surface predictor was a factor variable. Moisture too was closer to factor, however I did not consider the case of factorizing Moisture at this time, due to complex nature of data exploratory analysis and methods involved in examining the problem.

I then created a Cross validation data splits between train and test. I took one fold of train data from cross validation split and conducted a full linear regression and attempted to conduct a full data analysis using Linear Regression approach, diagnostics, model selection criteria, analysis and discussion about outliers, influential points and leverage. Additionally, as suggested by the authors from

[19] I conducted an Anova test and analyzed and gave a full pass over this 1 fold train dataset.

I then took entire train dataset and repeated cross validation over

(1) Linear Regression (lm())
(2) Lasso Regression $(glmnet(alpha = 1, lambda = best_lambda))$ with hyperparameter tuning.
(3) Ridge regression $(glmnet(alpha = 0, lambda = best_lambda))$ with hyperparameter tuning.
(4) Support vector regression using SVM.
(5) using glmnnet and neuralnet packages, I conducted Neural network solution for linear regression.

For each of the above approaches, I calculated metrics defined earlier in the paper. Results are discussed in the following section. Support Vector Machine and Neural networks solutions do not have likelihood function. GLMNET package limits using AIC, BIC default R package methods used. Hence I developed by referring to R tutorials from https://garthtarr.github.io/avfs/lab03.html.

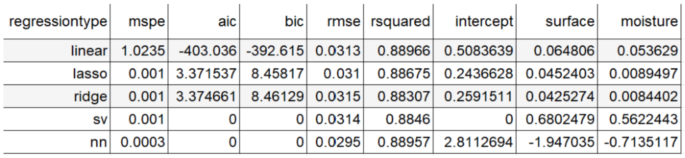
Since it is well known that response variable Static Friction Coefficient does hold true for Linear regression over Surface and Moisture content, it seemed simpler to interpret the results from Root Mean Squared error, MSPE.

Lastly, I implemented as well as adapted two different approaches for simple non-linear data i.e. XOR dataset to explain how all of the approaches compare. Neural networks are more adaptable between linear and non-linear data with better predictive power.

9 EVALUATION AND CONCLUSION

As part of the evaluation, despite the practical work experience over Machine learning models, I found Linear regression was much more explainable, interpretable and thus more relatable to an analyst. Comparing the slow progression of complexity from Lasso, Ridge, Support vector regression to Neural Networks, I found except for spending time to analyse the results, less explainable nature of the Neural Networks, it seems like less interpretable about how Neural Network with a hidden layers of 64 units and a final layer using Rectified Linear Unit (ReLU) has such unexplainable predictive power. However it seemed reasonable to use RELU that predicted and adapted to this specific dataset, with far less explainable features than that of Linear Regression in just one initial full pass over elaborate data analysis over just simulated data as well as the dataset.

The results from the data analysis materials provided here https://tinyurl.com/5n6es9tx are as follows: Generally each of the approach did get a good RSquared values suggesting variance was explained approximately 88%. Mean squared prediction error (MSPE) is ideally expected to be zero, however it was the least in case of Neural Networks regression model. The MSPE was higher in case of Linear regression, which needs to be further examined.



regressiontype	mspe	aic	bic	rmse	rsquared	intercept	surface	moisture
linear	1.0235	-403.036	-392.615	0.0313	0.88966	0.5083639	0.064806	0.053629
lasso	0.001	3.371537	8.45817	0.031	0.88675	0.2436628	0.0452403	0.0089497
ridge	0.001	3.374661	8.46129	0.0315	0.88307	0.2591511	0.0425274	0.0084402
sv	0.001	0	0	0.0314	0.8846	0	0.6802479	0.5622443
nn	0.0003	0	0	0.0295	0.88957	2.8112694	-1.947035	-0.7135117

10 WHAT COULD BE BETTER? AND FUTURE WORK

True coefficients are unknown, however Linear regression could be worked better by factorizing Moisture feature. I would like to continue working on this as an extension and work more to find out if best Linear fit with reduced MSPE solution is possible. As a next natural step, I would also like to find out more about ANOVA analysis and how to better adapt the dataset more towards good explanation since dataset is already known to satisfy Linearity assumptions as per [19].

11 PROJECT ARTIFACTS

All artifacts are placed under: https://tinyurl.com/5n6es9tx

Project HTML version of R Notebook for the dataset: https://tinyurl.com/3cejc28b

HTML version of R Notebook for the simulated data: https://tinyurl.com/2cz26s4y

Project work : R Notebook .ipynb version of the project and Data analysis https://tinyurl.com/yrbxzs4c

Project work : R Notebook .ipynb version of the simulated data and analysis: https://tinyurl.com/3t8ayc67

Project report without Programming part is: https://tinyurl.com/5n6es9tx

12 ACKNOWLEDGMENTS

Thank you so much to the Prof. Brian Zaharatos for guidance, support, feedback and discussion.

REFERENCES

[1] [n. d.]. The m of n rules and linear seperability. https://www.cis.upenn.edu/~danroth/Teaching/CS446-17/LectureNotesNew/intro/main.pdf

[2] 2005. https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/datopic.html

[3] Mariette Awad and Rahul Khanna. 2015. *Support Vector Regression.* Apress, Berkeley, CA, 67–80. https://doi.org/10.1007/978-1-4302-5990-9_4

[4] Rishabh Bhardwaj, Navonil Majumder, and Soujanya Poria. 2021. Investigating gender bias in bert. *Cognitive Computation* 13, 4 (2021), 1008–1018.

[5] Capobianco, Gilpin LaRosa, Xiang Sun, and Feldman. 2021. eXplainable AI approaches for debugging and diagnosis. https://neurips.cc/virtual/2021/workshop/21856. [Online; accessed 14-December-2021].

[6] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. 1983. Machine learning: A historical and methodological analysis. *AI Magazine* 4, 3 (1983), 69–69.

[7] Gabriele Ciravegna, Francesco Giannini, Marco Gori, Marco Maggini, and Stefano Melacci. 2020. Human-Driven FOL Explanations of Deep Learning.. In *IJCAI*. 2234–2240.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[9] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[10] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer.

[11] Farzad Jalilian Tabar, Rashid Gholami, and Ali Nejat Lorestani. 2011. Humidity effect on coefficient of static friction of rosemary and lavender by friction-electric meter. *Journal of Medicinal Herbs,* 2, 3 (2011), 187–191.

[12] Nicol Turner Lee. 2018. Detecting racial bias in algorithms and machine learning. *Journal of Information, Communication and Ethics in Society* (2018).

[13] L.E. Melkumova and S.Ya. Shatskikh. 2017. Comparing Ridge and LASSO estimators for data analysis. *Procedia Engineering* 201 (2017), 746–755. https://doi.org/10.1016/j.proeng.2017.09.615 3rd International Conference "Information Technology and Nanotechnology", ITNT-2017, 25-27 April 2017, Samara, Russia.

[14] Thomas M. Mitchell. 1997. *Machine Learning* (1 ed.). McGraw-Hill, Inc., USA.

[15] Christoph Molnar. 2019. *Interpretable Machine Learning*.

[16] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. 2018. Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and interpretable models in computer vision and machine learning*. Springer, 19–36.

[17] Stuart Russell, Daniel Dewey, and Max Tegmark. 2015. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine* 36, 4 (2015), 105–114.

[18] Andrew D Selbst and Solon Barocas. 2018. The intuitive appeal of explainable machines. *Fordham L. Rev.* 87 (2018), 1085.

[19] Seyed Mojtaba Shafaei, Arash Nourmohamadi-Moghadami, and Saadat Kamgar. 2017. Experimental analysis and modeling of frictional behavior of lavender flowers (Lavandula stoechas L.). *Journal of Applied Research on Medicinal and Aromatic Plants* 4 (2017), 5–11.

[20] Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the Black Box of Deep Neural Networks via Information. *CoRR* abs/1703.00810 (2017). arXiv:1703.00810 http://arxiv.org/abs/1703.00810

[21] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[22] Google's what-if tool. 2021. Visually probe the behavior of trained machine learning models, with minimal coding. https://pair-code.github.io/what-if-tool/index.html#demos. [Online;2021].