

University of Duisburg-Essen · Campus Duisburg

Faculty of Engineering

Chair of Mathematics for Engineers

Bachelor Thesis

For obtaining a Degree in

**B.Sc. Computer Engineering (Software Engineering)**

**Comparative Data Analysis using Lasso Regularization and Elastic Net**

**Moksha Jain**

Matrikel-Nr.: DS0302192300

from Duisburg

**Primary Examiner:** Prof. Dr. rer. nat. Johannes Gottschling

**Secondary Examiner:** Saad Alvi, M.Sc.

**Advisor:** Saad Alvi, M.Sc**.**

Duisburg, 18.03.2019

Kurzzusammenfassung

Abstract

**Versicherung an Eides Statt**

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

* dass ich die vorliegende Arbeit - mit Ausnahme der Anleitung durch die Betreuer
* selbstständig ohne fremde Hilfe angefertigt habe und
* dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
* dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
* dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ort, Datum) (Unterschrift)

Table of Contents

[Abstract III](#_Toc515671)

[Table of Contents V](#_Toc515672)

[Table of Figures VII](#_Toc515673)

[List of Tables VIII](#_Toc515674)

[Table of Listings IX](#_Toc515675)

[List of Abbreviations X](#_Toc515676)

[1 Introduction 1](#_Toc515677)

[1.1 Motivation 1](#_Toc515678)

[1.2 Problem Description 2](#_Toc515679)

[1.3 Structure of the Thesis 2](#_Toc515680)

[2 Basic Concepts 3](#_Toc515681)

[2.1 Data Pre-Processing 3](#_Toc515682)

[2.2 Feature Selection 4](#_Toc515683)

[2.3 Classification and Regression 6](#_Toc515684)

[2.4 Regression 7](#_Toc515685)

[2.4.1 Best-fit line and cost function 7](#_Toc515686)

[2.4.2 Linear Regression 8](#_Toc515687)

[2.4.3 Bias and Variance 9](#_Toc515688)

[2.5 Regularization 9](#_Toc515689)

[2.5.1 Ridge, LASSO and Elastic-Net Regression models 11](#_Toc515690)

[2.6 Cross Validation 12](#_Toc515691)

[2.6.1 K-fold Cross Validation 13](#_Toc515692)

[2.6.2 Cross validation in Embedded methods 13](#_Toc515693)

[2.7 Software Used 13](#_Toc515694)

[2.7.1 Python 13](#_Toc515695)

[2.7.2 PyQt 13](#_Toc515696)

[2.7.3 EIDOminer 14](#_Toc515697)

[3 Implementation 15](#_Toc515698)

[3.1 GUI 15](#_Toc515699)

[3.2 Controller 15](#_Toc515700)

[20 15](#_Toc515701)

[4 Experimentation 16](#_Toc515702)

[5 Summary 17](#_Toc515703)

Table of Figures

List of Tables

Table of Listings

List of Abbreviations

# Introduction

In today’s era of internet, information is the key tool in personalising the experience of any application at consumer as well as industry level. In the fields of social media, e-commerce, health care and medical diagnosis systems, various tech giants like Facebook, Google, Amazon etc. are pooling in their resources to use machine learning in achieving competent systems for optimizing user experience.

The advancements of such system results in a rapid growth of data. This brings in a challenge to store and manage the data efficiently. With an aim to achieve an effective way to maintain huge amounts of data, Machine learning techniques are used to infer patterns and extract knowledge which is representable to any given system. The downside of this is that redundant factors as well as incomplete data may be present in the data set. This results in a poor performance of the algorithm used.

Various techniques are developed in modern day Data Science to conquer such problems. These methods use statistics and mathematics to compare importance of various factors present in the data and use only the ones which are most significant. Such techniques are discussed and compared in the following thesis.

## Motivation

Data Science is the basis of knowledge representation of Artificial Intelligence and Machine Learning. It is important to provide good quality data to the algorithm, to reduce time and complexity required to deduce patterns and learn from them. To accomplish an adequate quality of data, it is important to take care of its repetitive and dysfunctional part.

Many modern-day algorithms using statistical methods focus on analysing and reducing the given input data to its core representation. The following thesis aims to compare these methods and suggest an optimal algorithm to use, which would result in a fast and accurate performance of the machine learning model.

## Problem Description

The following thesis mainly focuses on feature selection of given dataset using Lasso Regularization and Elastic Net.

The main problem encountered in this thesis is to choose a viable method for feature selection, use that on the given datasets and select most significant features from the dataset. The selected features (reduced dataset) are then given to EIDOminer to train specific machine learning models and test the accuracy.

To compare the efficiency of feature selection, the machine learning algorithms are also trained and tested using original dataset. The training, testing and validation error of this analysis is then compared to that of reduced dataset.

## Structure of the Thesis

# Basic Concepts

## Data Pre-Processing

Data pre-processing refers to methods applied to the data set before feeding it in the machine learning algorithms.

Data pre-processing is an approach to extract a clean data set from a given raw data. The collected raw data can come from different sources and is therefore not appropriate for analysis. If irrelevant or redundant information is present in the dataset, the training phase of the algorithm becomes slower and more complex. Data pre-processing methods helps to remove any irregularities and create a clean and reliable dataset. [1]

Data pre-processing methods include following categories [1,2]:

1. Data Cleaning

This process refers to methods that ‘clean’ the data by filling in empty values, smoothing noisy data, removing outliers and dealing with inconsistencies in the dataset. The machine learning algorithm is prone to overfitting, biasness, misclassification/false prediction if the dataset contains any irregularities.

1. Data Transformation

This involves methods to consolidate the dataset into forms which are appropriate for yielding the maximum output of the algorithm. This includes normalization (scaling attribute to small specific range), aggregation (applying aggregation operations) and generalisation (replacing with higher level concepts using hierarchies) of data.

1. Data Reduction

Data reduction methods result in different reduced representations of the dataset without compromising the integrity of the data. The idea behind this can be explained as reducing the number of attributes or dimensions from provided data. These representations are much smaller in volume but contain only the critical information.

One of the strategies to reduce data is, Dimension reduction where redundant attributes are detected and removed.

## Feature Selection

While applying machine learning algorithms on data with high number of variables, an important setback comes in picture called ‘curse of dimensionality’. It refers to the phenomena of data appearing much sparser in high dimensional space affecting the performance of algorithms designed for computing operations in lower number of dimensions. Higher the number of features, more are the chances to overfit the learning model which may affect the accuracy to predict on unseen testing data.

Dimensionality reduction is the state-of-the-art method for data reduction. It can be categorised into two main components:

Feature Extraction, this converts a high dimensional space to a low dimensional space. The features of new dataset created are usually linear or nonlinear combination of original dataset. Common methods for feature extraction are: Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Singular Value Decomposition etc. [3]

Feature Selection: Itis a process of selecting the most significant features out of given feature space. This is done by analysing the features present in the dataset and creating a subset of relevant features which represent the dataset most accurately.

Feature selection methods can be categorized in three distinct categories:

**Filter Methods:**

These methods are often used in pre-processing step. These are independent of the learning algorithms and depend purely on data to determine the importance of features. [3] They use statistical methods to score the variables based on their correlation to the resulting variable.

The variables with the highest scores are then selected and passed to the learning model for training. Various correlation methods used are LDA, Pearson, Chi-squared, and ANOVA.

These methods are very fast in computation and reduce the data volume significantly. They are also robust to overfitting(ExplainWhy???) which may result in better performance.

But they do not take multicollinearity between the independent variables into consideration. This has to be removed in a separate step before training the model.



Fig: Filter method flow diagram

**Wrapper Methods:**

These methods work according to a specific learning model. They evaluate the quality of selected features based on the performance of the set on the learning algorithm. It consists of mainly two steps: first, to select a subset of features and then it evaluates it. These steps repeat until the maximum performance or minimum error is achieved. The criteria for that could be set. In the flow diagram below (figure 2), whole data is provided to the Feature set Search. It uses wrapper methods to select the optimal subset. The selected subset is then evaluated using the learning algorithm. Based on this hypothesis, an evaluation is created and sent again to the set generation methods. Depending on how performance improves/reduced with each new variable in subset, it selects/rejects the variable from main feature set generated. As the criteria is achieved, the iteration stops and selected variables are provided as output set. The criteria here may depend on number of variables to be selected, or select all with maximum performance. [3]

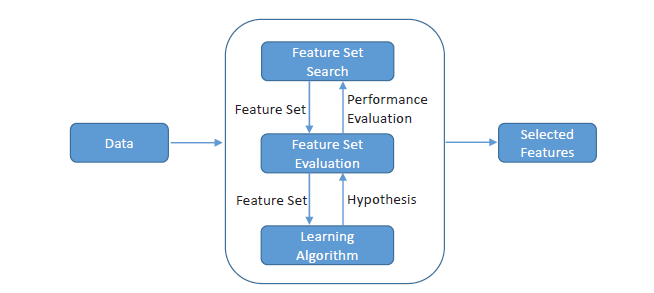


Fig: Wrapper method flow diagram

The two commonly used wrapper methods to search the subset are:

Forward Search, they start with empty subset and keep on adding new variables which enhance the performance.

Backward search, unlike to forward search, they start with the complete dataset provided and drop the ones which either scale down the performance or bring no change to it.

The problem faced with wrapper methods is for n features the required search space 2^n [3], which may take huge amount of time for higher values of n, increasing the complexity and the costing of algorithm, thus making it an inadequate option to be used for feature selection.

**Embedded method:**

These include qualities of both, filter and wrapper methods. Like wrapper methods, they embed the variable selection procedure using model training but don’t evaluate the features iteratively, eradicating the problem of exponential space search. And compared to filter methods, they focus on bias of learning algorithm and thus are optimal for learning [3].

Commonly used embedded methods are based on regularization models, which focuses on minimising the fitting error and making the feature coefficients to be zero in order to remove those features.

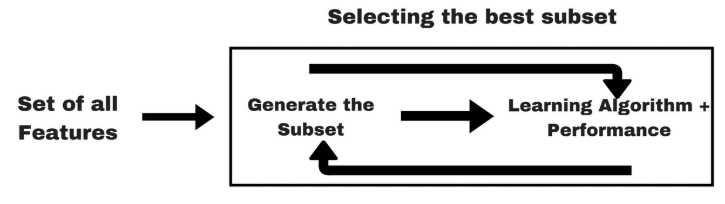


Fig: Embedded method flow diagram

## Classification and Regression

Predictive modelling is the art of developing a model that predicts results of new data without the knowledge of actual outputs.

These models can be described as a problem of mapping a mathematical function based on certain input variables (X) to output values (y).

Based on the type of outputs encountered, the prediction problems can be classified into two main categories:

1. Classification: Classification problems can be defined as assigning a category to the incoming input data based on a given list of classes. The classification algorithms learn and approximate a function based on a certain training data. For every new input data point, they analyse the independent variables according to the learnt function and assign the class based on the output of the function. The classes or the output values are discrete and finite in number.
2. Regression: Regression methods can be defined as predicting a function based on the given inputs(X) and using the function to calculate the output(y) of the new unknown data points. The main implementation of regression algorithm is for continuous values which are often quantities. Not all algorithms having regression in their name are regression algorithm, like Logistic regression. Its classification algorithm.

## Regression

Regression is statistical analysis used to predict relationship within variables. It helps predicting how a variable depends on other variables by developing a function for dependent (output/target) variable on one side and the independent variables with their dependencies (coefficients) on other side.

Business and banking sector, growth of any business, stock market could be considered example of real-life applications of regression.

### Best-fit line and cost function

Based on the regression function generated, a best fit line is created that fits the regression model. According to this line, the actual value must be close to approximated value in order to reduce the error. The distance between actual value and predicted value through line is called residual based on which the error is calculated.

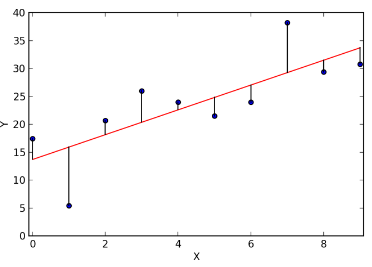


Fig: Error in Best fit line

Error is calculated in multiple ways:

Sum of residuals,(Y-f(x)), sums up all residuals. The problem encountered here is that the positive and negative residuals might cancel each other resulting in wrong error generation.

Sum of absolute value of residuals, (|Y-f(x)|), sums up absolute values of residuals which avoids the cancellation of values.

Sum of squared residuals, , sums up squared values of residuals. Here, the error is escalated to show a significant difference in larger and smaller error which helps selecting the best fit line.

Cost Function: It is the function used to define and calculate the error of the model. It can also be described as loss value of the model. Each error type has a different of cost function. For example, the cost function for squared residuals is:

Mean Squared Error (MSE) =

The main task is to minimize the cost function. This is done by using gradient descent. In this method, cost function is calculated at each point iteratively to find the minimum value of it.

### Linear Regression

Types of regression depend upon type of dependent variable, number of independent variable and shape of regression line. For example, Polynomial regression where the power of independent variable is more than one. Thus, equation formed in this case would be y=a\*x^2+b. the best fit regression line in this case is a curve. The regression type used here is Linear Regression. [4]

The simplest linear model for regression is one that involves a linear combination of

the input variables

*y*(**x***,***w**) = *w*0 + *w*1*x*1 + *. . .* + *wDxD*

where **x** = (*x*1*, . . . , xD*)T. This is often simply known as *linear regression*. The key property of this model is that it is a linear function of the parameters *w*0*, . . . , wD*. [5]

A simple linear regression equation is:

y = w0 + x\*w1

A linear regression always gives a straight line where w0 id the y intercept and w1 is slope of the graph which tells weight of the variable.

### Bias and Variance

Bias and Variance are the two main types of error encountered in any machine learning algorithm. In order to fit data perfectly and to have consistent predictions, it is very important to understand bias and variance.

Bias is the error generated due to assumptions made in the model. It is directly proportional to error between predicted and actual value. It can be mathematically written as E(Y´-Y), where Y´ is predicted function and Y is the function to be predicted.

Variance is the variability in results predicted by model. It is the difference in fits between data sets. If a model fits training set well but not the testing set, it is said to have high variance.

Bias and variance are related to common terms in machine learning, overfitting and underfitting. Higher variance and lower bias leads to overfitting which means model fits very well for a particular dataset but fails to fit other model. It is often caused by a complicated model. On the other hand, lower variance and high bias could cause underfitting by not capturing the data properly which is caused by a simpler model.

The underfitting problem can be resolved by increasing complexity of the model by adding new parameters. But to solve overfitting problem, complexity needs to be reduced. One way to do so is to regularize it.

## Regularization

Regularization is an important concept in the world of machine learning. It can in general be defined as making things regular. In the three plots shown below, each line tries to fit the data.

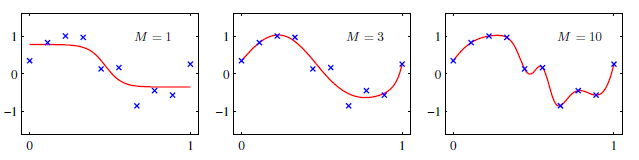


Fig: Regularization, explain all 3 plots

In The first plot with degree M=1, the model underfits the set producing a large error. Going to higher degree, with M=10, the model overfits as now its cost function may give 0 for training set but due to so many fluctuations the cost function for unseen (testing set) may give a higher error value. Comparing to both plots in the left and right, the graph with degree M=3 fits the data just perfectly and since it seems to be stable, with not many fluctuatuins, it fits the unseen data well too. [5]

Most models encountered have the problem of overfitting. In order to solve it, one uses regularization. Thus, via regularization the highly inconsistent model can be regularized in a smooth graph.

A model is regularized by adding an additional term to its cost function. This term helps model reducing the fluctuations by avoiding the coefficients to take extreme values. It weights the coefficients and ones with not so higher value signifies lower importance and thus can be dropped. Therefore, regularization not only helps in normalizing rthe model but also reducing the features based on their importance to it. [5]

To understand it more clearly, example of linear regression is considered.

This function describes error function for linear regression where is the real value and ,the predicted value, is subtracted from it. Here, n is number of data points considered in training example and p is number of parameters or features. is the data point and is the parameter being learnt. Explain proprerly…language

To avoid overfittimg in this model, regularization term is added to the function. The resulting function is,

Here absolute value of each parameter is added to the sum, which contributes to overall sum. The idea behind adding this term is to perform badly as cost function to reduce overfitting. Since we are taking minimum of the whole total value the higher the sum is the lower is the performance. In practice the tuning parameter , that controls the strength of the penalty, assume a great importance. Indeed when is sufficiently large then coefficients are forced to be exactly equal to zero, this way dimensionality can be reduced. The larger is the parameter the more number of coefficients are

shrunk to zero. On the other hand, if = 0 we have an OLS (Ordinary

Least Square) Regression. This phenomena is often reffered to as weight decay in machine learning world as it encourages weitght value to decay towards zero.

There are two main approaches to regularize a function. [6]

L1 Loss: the ablsolute value of weight is added here.

L2 Loss: the squared value of weight is considered here.

These loss functions are used in Lasso and Ridge regression which are the algorithms used to select features from a given set.

### Ridge, LASSO and Elastic-Net Regression models

Regression models are used for many purposes. For instance, analysis of variance

(ANOVA), parameters estimation, prediction and variable selection. [6]

These use similar concept of regularization. **Ridge regression** uses L2 Loss function to detect the error value. It quantifies the weight by squaring the values which brings a clear difference in parameters and their importance.

On the other hand, **LASSO** **Regression** uses L1 Loss function that adds up the absolute values of parameters.

When LASSO optimization problems are minimised, some variables are shrunk to zero, i.e. weight, = 0, depending on value of (value of decrease with rise in). In this way the features with coefficient equal to zero are

excluded from the model. For this reason, LASSO is a powerful method for

feature selection while other methods (e.g. Ridge Regression) are not. [6]

But LASSO itself has some limitations:

In small-n-large-p dataset the LASSO selects at most n variables before it saturates.

If there are grouped variables (highly correlated between each other) LASSO tends to select one variable from each group ignoring the others. [6]

**Elastic Net** **Regression** overcomes LASSO limitations using a combination of LASSO and Ridge Regression methods.

Where, and are two regularization parameters.

Adding a quadratic part to the penalty, Elastic Net removes the limitation on the selected variables number and stabilize the selection from grouped variables.

Thus, Elastic Net which is a convex combination of Ridge and LASSO Regression is theoretically assumed to perform better than both individual methods.

The size of penalty terms () can be tuned using cross-validation method in order to find the model’s best fit. [7]

In python, the term lamda used for mathematical notation of tuning parameter is denoted using alpha.

The equation for Lasso and LassoCV is:

(1 / (2 \* n\_samples)) \* ||y - Xw||^2\_2 + alpha \* ||w||\_1

For Elastic Net the function is:

1 / (2 \* n\_samples) \* ||y - Xw||^2\_2

+ alpha \* l1\_ratio \* ||w||\_1

+ 0.5 \* alpha \* (1 - l1\_ratio) \* ||w||^2\_2

Alpha is the constant that multiplies the L1 term. Its default value is 1.0. The lower the alpha goes, the result tends to get equivalent to the ordinary least squares.

The CV used in LassoCV and ElasticNetCV is an integer value that implies to use cross-validation for an iterable yielding splits as array of indices. If the parameter is not provided the default value is taken as 3-fold-cross-validation. [7]

## Cross Validation

A common approach in supervised learning is to break training set in learning and validation set. this is done so that model is trained from training set, validated from validation set and if validation is successful, it could be evaluated finally via testing set. But this partitioning of data in three parts drops the performance drastically by reducing number of samples used to train the model. To solve this problem, cross validation method is used. [8]

### K-fold Cross Validation

K-fold cross validation is basic approach towards cross validation. Here, the training set is split in k smaller sets. The following procedure is then followed for each k-fold:

Firstly, the Model is trained using k-1 folds as training data and then the model is validated on remaining fold/part of data. The last part of set is treated as test set to compute the performance for k-1 folds.

The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop. This approach can be computationally expensive but does not waste too much data (as is the case when fixing an arbitrary validation set), which is a major advantage in problems such as inverse inference where the number of samples is very small. [8]

### Using Cross Validation to calculate tuning parameter

Lamda is the tuning parameter used in Lasso and Ridge regression. Higher the value of lamda is the flatter is the curve of regression.

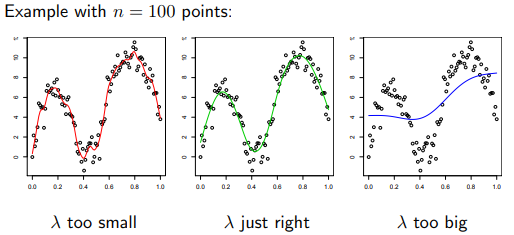


Fig: curve change with lamda

The calculation of tuning parameter in general can be understood as follows:

For each tuning parameter in equation,

In this equation, is calculated on the training set and error is calculated on the validation set. For each , the average error is computed over all the K folds.

The CV() curve is made though this and is chosen which minimizes the curve.

= argmin  ∈{θ1,...θm} CV()

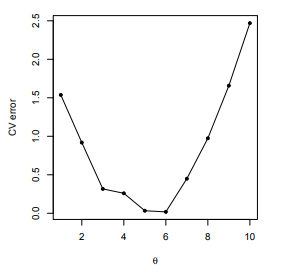


Fig: Θ vs CV error curve

## Software Used

### Python

Python is a high-level programming language that uses dynamic type system and automatic memory management. Multiple programming paradigms like object oriented, functional, procedural are supported by the language.

The IDE used here to use python is PyCharm by JetBrains. Python uses Scikit-learn module to merge classic machine learning algorithms with Python packages such as NumPy, SciPy, matplotlib.

### PyQt

PyQt is one of the most popular Python bindings for Qt cross-platform C++ framework. It was developed by Riverbank Computing Limited. PyQt provides two bindings: Qt4 and Qt5. PyQt5, version used here is built on Qt5.x.

In general, it could be considered as platform used to develop GUI especially for Python implementations.

### EIDOminer

EIDOminer is a prediction software, based on various machine learning algorithms which gives an extensive analysis and the quality of data before it if given as input to other mathematical tools. The core to this software is Intelligent Analyse manager (IAM). [9]

# Implementation

The core implementation for this Thesis consist of a GUI application programmed using Python. Python was chosen in this Thesis as it includes various inbuilt libraries for data visualization and processing. Also it provides support for making interactive Graphical user interfaces using PyQt or Tkinter.

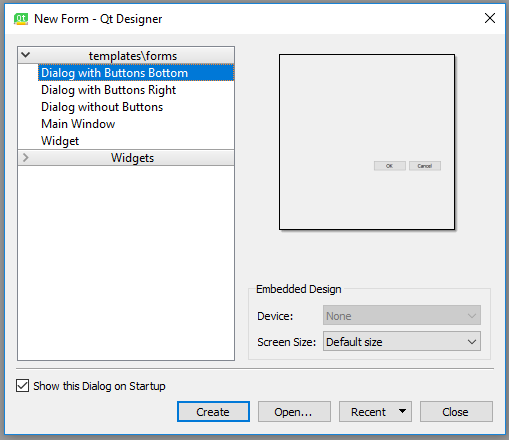
The Application is programmed using Object Oriented Programming concepts in both GUI and Controller class which are explained later in this chapter.

Python uses Scikit-Learn library to implement machine learning algorithms. Scikit-learn is an open source and is simple and efficient tool used for data mining and data analysis. It features various Regression, Dimensionality Reduction, Pre-Processing, Model Selection algorithms such as PCA, cross validation, SVM, k-means etc. This library is built on python numerical and scientific libraries NumPy, SciPy and Matplotlib.

NumPy consists of functionalities for handling large multidimensional arrays and matrices. SciPy contains modules for scientific and technical computing such as linear algebra, signal pre-processing and optimization, integration etc. Matplotlib provides features for mathematical visualizations for data like histograms, charts etc.

## GUI

The interface which enables user to interact with the core application or the algorithm behind it is referred to as Graphical User Interface. The class defined in the project for GUI application is named as LoadFilesWindow. The file was originally developed using PyQt5 in QtDesigner in a .ui format, later was converted in .py form to be able to be used in PyCharm.



### Design of GUI

A Dialog box was selected as the face of the application. Dialog box is a compact way to design an application with minimal features. The lineEdit fields used contain the paths to the training and testing files. Two browse buttons were used to provide interface for selecting the files. Using the push button for Browse, user can select the file for train and test data. Path of the selected file was then added as the text in lineEdit box for respective file. Next, User must also provide a method to be used for feature selection. A list of radio buttons, each indicating one method, were used as multiple methods cannot be used at the same time. The user must select atleast one method in order to proceed and enable the OK button.

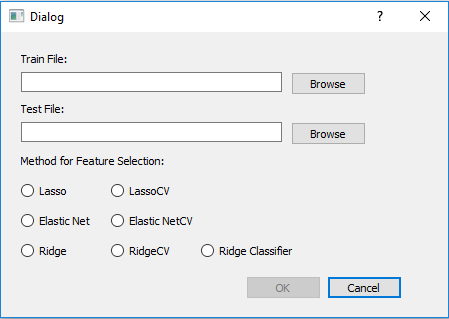


Fig: Dialog Box for user input

After selecting a method and clicking OK, it goes to function called ToSelectFeature(), where the DialogBox interacts with the Controller class. At first, the files are sent to controller. Now Feature set in Training Set (XTrain) and output variable (yTrain) are extracted by getting the returned arrays from function TrainDataRead() in controller which reads the Training set and returns the variable and result set respectively. The same is followed for testing data set. The Main dialog box closes after this occurs.



A dialog box to save the new file is then generated using QFileDialog. it asks for the path and name of the file and saves it with .csv extension.

Depending on which radio button is selected by user, the method is initialized in the controller. For each type of method a function is defined in controller class which is called for specific type of selection using the controller object.



The function selectFeatures() defined in controller class is called that sends in all the datasets and the path to the result file and the selected method as parameters.



The GUI and Controller works parallel to each other for this application.

## Controller

A controller can be understood as the brain of the code. It contains the required data and the implementations of all the necessary functions from where the output is generated. NumPy and Pandas are imported to handle the datasets in form of arrays and matrices. Scikit-learn library was used in order to use SelectFromModel, which is a Meta-transformer for selecting features based on importance weights. Sklearn.linear\_model also provides options for various embedded methods for feature selection such as ElasticNet, Lasso, Ridge etc.

The controller consists of multiple methods. The class defined by its constructor consists of various variable definitions as they are required and initialised in different methods and need to be accessible in other functions of the class. These variable and their purpose are as follows:

Traindata, Testdata: stores training and testing data respectively. Initialised as none.

TrainFilename, TestFilename: stores file names of train and test data respectively. Initialised using values sent by GUI as parameter to constructor.

TraindataSet\_Header: stores the column names including feature names and the result name. initialised none.

MethodUsed: method selected by user for feature selection. initialised none.



To read the data from the provided testing data and training data CSV file, two methods, namely TrainDataRead() and TestDataRead(), read the CSV file using pandas and convert it into data frames which can be easily manipulated and interpreted in Python.



The delimiter was set as ‘;’ which means the values in the set were separated through a semi-colon ( ; ) . Since it is difficult to use mathematical operations on set defined by pandas, the set was then converted to numpy data array. TraindataSet\_Header is also updated using values from Traindata.

As python uses the English format for numbers (using (.) for decimal), the sets using German convention were then converted to English by replacing (,) by (.). Once the complete set is transformed, it is divided in feature and the result set as the feature selection algorithm is applied only on feature set. The complete data set is again converted in pandas for feature use. This Function returns Feature and Result set as its output.



The same procedure was repeated in third method of the class which is defined for reading and operating the test data. This function is named TestDataRead(), it as well returns the Feature and Result set after converting it in English format.

Then the methods for initialising the variable methodUsed are defined where the variable is assigned the method it would use for feature selection.

Then the methods for initialising the variable methodUsed are defined where the variable is assigned the method it would use for feature selection. Since there are various types of methods , a different method is defined for initializing the chosen method which is called in class GUI. These methods include Lasso, LassoCV, ElasticNet, ElasticNetCV, Ridge, RidgeCV, RidgeClassifier. The methods are used in the default settings with cv provided as 10 for using 10- fold cross validation as default.

In method selectFeatures, the task of selecting features from the data set takes place and the new file containing new data set is generated as output. The function takes the feature and result sets of train and test data with the path where the new file has to be saved as its parameters.

Firstly, by providing methodUsed in SelectFromModel the method to be used for feature selection is given to the model. The threshold parameter refers to value to use for feature selection. Features whose importance is greater or equal are kept while the others are discarded.Then feature set and result set of training data are provided to fit the model. Here algorithm weights the importance of features in set TrainX based on the result in TrainY. Depending on the significance features are selected through which the resulting set can be extracted using transform. Only TrainX set is used as features are selected only from this set.

Based on index numbers of columns selected from training set, their name is determined. The similar columns are extracted from testing set as well using iloc.

The parameter, feature\_idxn is array containing true at all indices where the column is selected and false where not. It also has the last column as false in order to keep the array size the same as that of the Testdata set.

The result header is generated that contains the headings of the selected variables. The heading from last column, the result name is appended to the header.

The final train and test data sets are first created individually by merging the new features set with result set. This is done using numpy. After these sets are ready, to get the final set its important to concatenate these sets. As the concatenation is done using pandas, the final train and test sets are converted to pandas data frames and then concatenated.

Since pandas add a row of 0 in the end of frame in some cases, that is taken care of by removing such row in case present as it can cause problem in predictions. The final data frame is then exported to a csv file which contains semi-colon (;) as separator and the Result header generated earlier as the header for the file.



Editions: Separate files for test and train to compare results

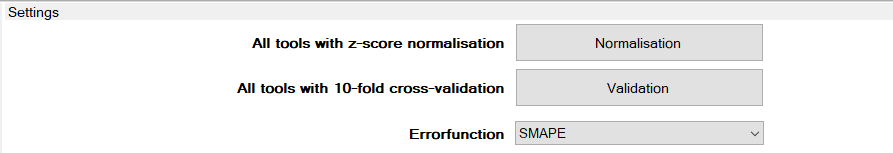
Changes in GUI: 2 save file boxes, send val to controller

Controller: donot concat, 2 files made. Language: eng-ger

# Experimentation

In order to design an output from the application presented above, various application runs were conducted. They were mainly based on four different data sets named as: C15kw, Elongation, KE\_Final and Cars. For each set, an initial analysis was generated in EIDOlerner followed by another analysis generated on the dataset created by application after feature selection process from original data set.

At first, the original data is fed to EIDOlerner. The analysis is created using 10 percent of data as test data. No pre-processing tools were selected as data is considered to be already pre-processed. Neural Network and SVM were selected as prediction tools to measure the performance. The analysis window is appears after that. To get better prediction analysis, k-fold cross validation as the validation method, SMAPE as error function and z-score as the normalisation method are selected.



To create the analysis, it was important to train the models with data. With neural network and Support Vector Machine, the model was trained by providing various values for parameters. To check if a training method is selected or rejected, error parameter is given. The error parameter was set as 0.1 as default for both methods, implying if error is less than 0.1 the method is considered to be good.

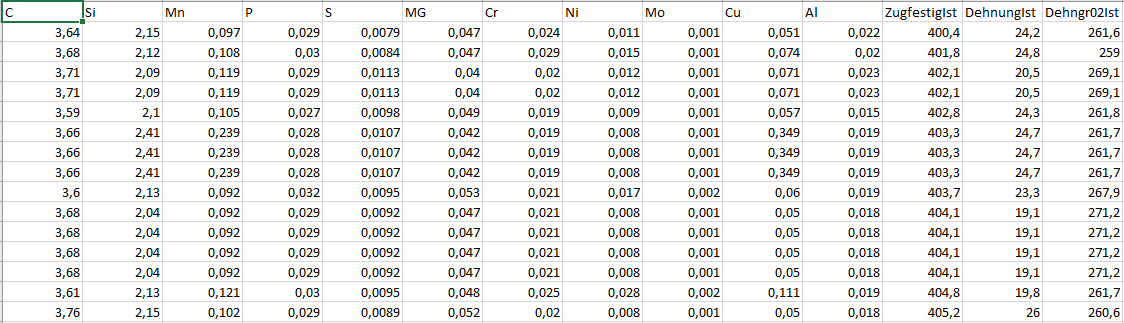
Additionally, for neural networks number of cycles was changed from 500 to 2000 as accuracy increased with higher number of cycles. The data was not shuffled and polar sigmoid was used as the activation function. For SVM, the Kernel was set to Gaussian with sigma value of 10000 which is set as default. Training and Validation error were generated after the model was trained. These errors are treated as the performance measure for each method. This analysis was saved as the main file and was used later for comparison.

Now from the supervisor, the training data (named as original data set) and the test data is exported in an Excel file. the data sets are first converted to .csv form and then loaded in the application. The application does feature selection based on method selected and the parameters provided. The parameters could not be set by the user in GUI but only be set in the controller class while calling the function. After the process run the two new .csv files are generated for training and test data. For different values of threshold and alpha with each of feature selection methods, various files were created and were compared in EIDOlerner. In each analysis, zero percent test data is separated as resulting in only training and validation error has to be calculated from the same set of values. Later after the error is calculated, the test data file is loaded in supervisor and testing error could be seen there. The values where minimum error was found, was then trained 10 times and the average of all values is given in the tables for each set. To ease the comparison within all feature selection methods, a particular set of parameters is chosen for all methods. A brief explanation of how changes occur in feature selection based on parameters is given as well.

Based on the procedure explained above, the experimentation was performed on the data sets. The comparative analysis for each set was done and is discussed below.

## KE\_Final

This is a continuous dataset with 180 rows and 14 columns. The 11 independent varibales are considered as features and the three dependent variable is the result variable which depends on independent ones. Variable ZugfestigIst, DehungIst and Dehgr02Ist depends on features C, Si, Mn, P, S, MG, Cr, Ni, Mo, Cu, Al. From the result variable the type of set is known that is Regression in this case. The small preview of set is as follows:



Data set for KE\_Final

To calculate and to compare the exact accuracy, the set was at first divided in three sets where each dependent variable treated as only result variable. Thus 3 files were created each with only one result variable and 11 features and the complete procedure was performed on all 3 individually.

The file with variable ZugfestigIst was used at first in EIDOlerner.

1. KE\_Final\_ZugfestigIst

After the test and train data are taken from supervisor, the size of training data is 162 x 11 and 18 x 11 for test data. After comparing the analysis using various values of threshold and alpha, the least errors were found as following for threshold 0.5 and alpha = 0.3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KE\_Final\_ZugfestigIst | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 11 | 0.0085 | 0.0086 | 0.0238 | 0.0092 | 0.0011 | 0.0372 |
| Lasso | 3 | 0.0147 | 0.0129 | 0.0145 | 0.0224 | 0.0184 | 0.0219 |
| LassoCV | 3 | 0.0142 | 0.0091 | 0.0136 | 0.0186 | 0.017 | 0.0184 |
| ElasticNet | 6 | 0.0113 | 0.0079 | 0.0102 | 0.0109 | 0.0043 | 0.0109 |
| ElasticNetCV | 8 | 0.0129 | 0.0167 | 0.0098 | 0.0095 | 0.0122 | 0.0133 |
| RidgeClassifier | 0 | - | - | - | - | - | - |

In the table above, it can be seen that accuracy decreases with reduced number of features. But when ElasticNetCV select 8 features, the error rate rises compared to Elastic Net which selects 6 features. For Lasso and ElasticNet, when alpha was set as its default value 1.0, the number of selected features reduced. The threshold was tested with 2 other values 0.25 and 0.75, where all methods didn’t show any major difference for both values and selected same features in both cases. This means that the weight of each selected variable was high enough to pass though larger threshold as well. Comparing the feature selection in methods and their respective cross validation method, Lasso and Lasso CV results in same number of features with all tested values of threshold and alpha. ElasticNetCV always selects 8 and Elastic Net choose 6 features for alpha 0.6 and 4 features for alpha 0.4. Thus, ElasticNetCV in this case selects higher number of features than ElasticNet. Ridge classifier does not select any feature here as it doesn’t perform well for regression methods.

For this set, Elastic Net can be the option as the method selects optimal number of features with reduced error rate.

The important features for variable ZugfestigIst can be classified in following sets according to different selection methods: (C,Mn,Cu)->(Si,Mg,Ni)->(Cr,Al)

1. KE\_Final\_DehungIst

The selected value for threshold and alpha for this set are 0.5 and 0.3 respectively.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KE\_Final\_DehungIst | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 11 | 0.0335 | 0.0219 | 0.0673 | 0.0312 | 0.0523 | 0.0928 |
| Lasso | 2 | 0.0488 | 0.077 | 0.0519 | 0.0595 | 0.0721 | 0.0609 |
| LassoCV | 4 | 0.0408 | 0.0263 | 0.0323 | 0.0342 | 0.017 | 0.0392 |
| ElasticNet | 3 | 0.0401 | 0.0437 | 0.0405 | 0.0417 | 0.0386 | 0.0411 |
| ElasticNetCV | 7 | 0.0391 | 0.0458 | 0.0313 | 0.0315 | 0.0304 | 0.0398 |

In the error table for KE\_Final\_DehungIst, it can be clearly seen that lasso doesn’t perform well in this case. As it selects 2 features from 11 giving a very low accuracy compared to other methods. ElasticNetCV on other hand chooses 7 features and on comparative lower error rate. For this set both cross validation methods select larger number of features compared to their normal methods. Again, with increasing value of alpha to 1.0, Lasso and Elastic Net selects only 1 feature and the threshold from 0.05 to 0.75 gives the same result. This shows the weight of each selected variable is effective enough. After 0.75 threshold, the number of features reduces and so does accuracy. Thus, a middle value of 0.5 is considered ideal for the case and ElasticNetCV can be the optimal way to select features for this set.

The important features for variable DehungIst can be classified in following sets according to different selection methods: Cu->C->Mn->Si->(Cr,Al,Ni).

1. KE\_Final\_Dehngr02Ist

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KE\_Final\_ Dehngr02Ist | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 11 | 0.0093 | 0.0.076 | 0.0319 | 0.0118 | 0.0035 | 0.0379 |
| Lasso | 8 | 0.0111 | 0.0096 | 0.0066 | 0.0066 | 0.0067 | 0.011 |
| LassoCV | 4 | 0.0201 | 0.0151 | 0.0126 | 0.0131 | 0.0086 | 0.0195 |
| ElasticNet | 9 | 0.0109 | 0.0094 | 0.008 | 0.0086 | 0.0035 | 0.0107 |
| ElasticNetCV | 8 | 0.0121 | 0.0105 | 0.0059 | 0.0078 | 0.0078 | 0.0099 |

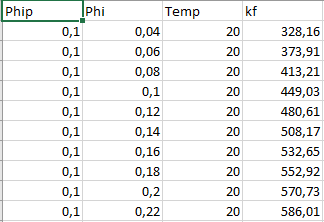
In the set KE\_final with Dehngr02Ist as result column, the error table is as above. Here the features are selected using threshold of 0.05 and alpha as 0.01. For this set number of features selected vary a lot with change in the value of parameters especially when using methods Lasso and ElasticNet. Lasso with alpha 0.3 selects 4 features and it doubles with alpha 0.01. ElasticNet accepts 8 features with threshold 0.05 and alpha 0.1. With same alpha and taking 0.15 as threshold, the number of selected features reduces to 6 and further reducing to 5 with alpha as 0.5. this difference gives a brief idea of weights of the features. With all values of threshold within 0.05 and 0.5 LassoCV selects 4 features and ElasticNetCV selects 8 features. In this case cross validation methods selects lesser features. Although all methods perform good for this set, but ElasticNet can be considered a better method as it weights the variables in distinct and diverse manner.

The important features for variable Dehngr02Ist can be classified in following sets according to different selection methods: (C,Si,Mn,Cu)->(Mg,Ni)->(Al,Cr)->S.

Now, considering the priorities of features selected for all result variables, the priority for set KE\_Final can be drawn as : Cu->C->Mn->Si->Ni->Mg->(Cr,Al).

## C15kw

C15kw is a continuous dataset consisting of 3 independent variables Phip, Phi and Temp and one dependent variable kf. The complete set has 1248 rows out of which 1123 are taken for training set and remaining for testing set. The dataset looks like:



Taking the parameters threshold and alpha as 0.5 each, led to following results.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C15kw | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 3 | 0.0212 | 0.0222 | 0.0226 | 0.0426 | 0.0476 | 0.0441 |
| Lasso | 3 | 0.0215 | 0.0229 | 0.0216 | 0.044 | 0.0415 | 0.0438 |
| LassoCV | 2 | 0.0693 | 0.0677 | 0.0688 | 0.0805 | 0.0749 | 0.0788 |
| ElasticNet | 3 | 0.0236 | 0.0212 | 0.0233 | 0.0418 | 0.0373 | 0.0414 |
| ElasticNetCV | 2 | 0.0721 | 0.0605 | 0.0659 | 0.0778 | 0.0678 | 0.0909 |

Based on the errors observed in table above, it can be said that all three features are very important to the set. As with reduced feature, the accuracy drops significantly. For Lasso and ElasticNet, using the threshold 0.05 to 0.5 accepts all features. After rising threshold to 0.75, 2 features (Phip and Phi) are selected resulting in unacceptable error which is higher than 0.1. further with 0.95, only one feature is selected. Thus, weight of all features is higher than 0.5 according to Lasso and ElasticNet. LassoCV and ElasticNetCV on the other hand chooses only 2 variables (Phip and Temp) even with very low threshold value of 0.05. This means, according to these methods ‘phi’ is assigned very less weight resulting in its elimination. Since the reduction of this feature brings hike in the error, this is not very optimal.

Lasso and ElasticNet perform equally well for this set but since they do not select any here they can not be considered ideal. Based on these observations, the priorities of the features can be deducted as: Phip->Temp->Phi.

## Elongation

Elongation is a set consisting of 17 columns for independent variables such as C, SI, MN, P, S etc. and one for dependent variable Elong. It has 385 rows and 347 from it are used for training and validation set.

## 

The following error were calculated from features selected using

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Elongation | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 6 | 0.0853 | 0.075 | 0.104 | 0.0043 | 0.0125 | 0.0347 |
| Lasso | 0 | - | - | - | - | - | - |
| LassoCV | 2 | 0.2258 | 0.25 | 0.1676 | 0.2237 | 0.2625 | 0.2428 |
| Elastic Net | 0 | - | - | - | - | - | - |
| Elastic NetCV | 2 | 0.2301 | 0.2357 | 0.2312 | 0.2237 | 0.2062 | 0.2428 |

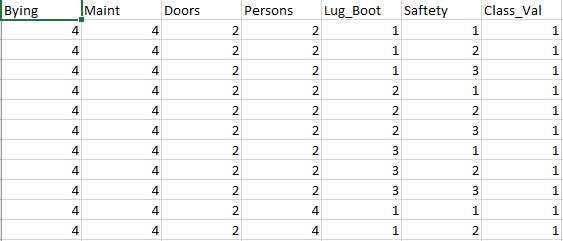
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cars | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 17 | 0.0194 | 0.0162 | 0.0372 | 0.0097 | 0.0114 | 0.0547 |
| Lasso | 3 | 0.0227 | 0.0229 | 0.0208 | 0.0211 | 0.0183 | 0.0227 |
| LassoCV | 3 | 0.0215 | 0.0228 | 0.0219 | 0.0201 | 0.0190 | 0.0230 |
| ElasticNet | 4 | 0.0218 | 0.0218 | 0.0206 | 0.0209 | 0.0185 | 0.0218 |
| ElasticNetCV | 3 | 0.0227 | 0.0227 | 0.0208 | 0.0271 | 0.0267 | 0.0227 |
| RidgeClassifier | 0 | - | - | - | - | - | - |

Different number of features are selected when alpha is changed.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| KE\_Final\_ Dehgr02Ist | alpha | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Lasso | 0.5 | 2 | 0.0097 | 0.0056 | 0.0086 | 0.011 | 0.0078 | 0.0121 |
| Lasso | 0.3 | 3 | 0.0085 | 0.0088 | 0.0057 | 0.0135 | 0.0133 | 0.0157 |
| Elastic Net | 0.5 | 5 | 0.0073 | 0.0046 | 0.0088 | 0.011 | 0.0084 | 0.0201 |
| Elastic Net | 0.3 | 5 | 0.0067 | 0.0074 | 0.0109 | 0.0077 | 0.0064 | 0.0157 |

# Cars

This is a classification type of dataset consisting of seven columns including six features and one result column, and 1728 rows. The features are Buying, Maint, Doors, Persons, Lug\_Boot and Saftety and result name is Class\_Val. The example of the set is given below.



After the data is divided in testing and training set, training set has 1555 and testing 173 rows. Following accuracies are calculated using different methods:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cars | No. of features | Neural Network | | | SVM | | |
| Training | Validation | Testdata | Training | Validation | Testdata |
| Original | 6 | 0.0853 | 0.075 | 0.104 | 0.0043 | 0.0125 | 0.0347 |
| Lasso | 0 | - | - | - | - | - | - |
| LassoCV | 2 | 0.2258 | 0.25 | 0.1676 | 0.2237 | 0.2625 | 0.2428 |
| Elastic Net | 0 | - | - | - | - | - | - |
| Elastic NetCV | 2 | 0.2301 | 0.2357 | 0.2312 | 0.2237 | 0.2062 | 0.2428 |
| Ridge Classifier | 5 | 0.0803 | 0.0812 | 0.1098 | 0.0366 | 0.0625 | 0.0578 |

From the table, it can be clearly seen that feature selection methods based on linear regression do not perform well on multiclass set as Lasso and Elastic Net selected no features. The methods using cross validation on the other hand selected few variables but has very less accuracy. To compare SVM and neural network, SVM performs better in this case as for original data and for Ridge Classifier it gives better accuracy and low biasness. LassoCV and ElasticNetCV performs almost equivalent in this case. They select equal number of features and gives similar accuracy with both models.

# Summary

### Cross validation reduces number of features

### Beneficial for larger sets. Limited features selected from huge sets am

### and the ratio of error increase is not a lot.