Prediction model of Colour Dryback

Padhi, Mayank

Award date:
2021

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Data Mining Research Group

Prediction model of Colour Dryback

Graduation Project

Mayank Padhi (1432788)

Supervisors:
prof. dr. Mykola Pechnizkiy
Fariba Safari (CPP)
Vishnu TV

Version 1.0

Eindhoven, Tuesday 10th August, 2021

Abstract

The problem of dry back has been around in the print industry for a very long time. Hence, with this project, we try to solve this problem partially, which means we do not eliminate the dry back but predict it. When a new type paper is used in a printer for the first time, one needs to make a color profile for calibration. The use of wet print results in inaccurate color profile and waiting for hours (or days) for color to dry costs time. The phenomena of dry back is dependent on various factors ranging from properties of the ink, the paper and settings of the printer. This scope of this report is limited to study the properties of the coated papers as we keep the keep the other factors as constants.

In this report we study the factors that affect the dry back. This study comprises of smaller individual studies. The impact of weight in the dry back is studied by considering a medium and taking different weight categories into consideration. The study revealed that the weight does not have a significant impact on drying when compared to the effect of the actual color i.e., the CMYK values. However, the impact is comparable to that of the media characteristics. Further, we study the media characteristics such as thickness, brightness, whiteness, bulk, opacity, gloss percentage etc. and try to study their impact. The results suggest that some of these characteristics have a more impact than others. Then, we use some of these factors to make regression models that predict the colors after they have dried. We study the previous approaches made to predict the dry back and later a range of regression algorithms such as the neural networks, tree-based ensemble learning algorithms and support vector regression algorithm.

The main contribution of this work is the study of the properties of coated papers and the study of their impact on the dry back. The regression models used in the past were limited to a few features for prediction. They only included the color aspect of drying ($CMYK$ and $L*a*b*$). This limited the consistent performance of their model to a certain paper types used in the training data. With this work, we have included the paper aspect to the prediction model which not only improved the accuracy but also expanded the range of the media types. The results from the regression model suggest that the predicted colors can be indeed used for making the color profile for a new paper type for a set of regions. However, the scope of the data used is limited to certain regions (EU and US). Hence, the drying behavior of coated media from other regions is yet to be determined.

Preface

My attachment with the field data science started during my bachelor degree in Computer Science. I used to get fascinated how the statistics and statistical learning theory had such huge real applications which initially did not seem intuitive. This laid the foundation and introduced me to the world of machine learning. This inspired me to go for higher studies and pursue a career in this field. So, this is how I ended up at the Eindhoven University of Technology. This journey was not exactly the way I (or anyone) anticipated but it was an amazing one, nonetheless. I take this opportunity to thank a few people who made it possible in these extraordinary times.

I would like to thank my mom and dad who supported me and kept my spirits up all this while. Mona, thank you for being a wonderful sister and a sparring partner. I would also like to thank all my friends in the Netherlands, India and elsewhere for always looking out for me.

Additionally, I would like to thank my supervisor at CPP, Fariba for her constant guidance, advise and feedbacks. I would also like to thank the color experts, media experts, data scientists, staff working with Niagara at CEC and members of the color management team. The project could stay on schedule despite the challenges posed by the pandemic because I was able to borrow their expertise at any time. This project would not have been possible without their inputs, feedbacks and constructive criticism. Further, I would like to thank my supervisors from TU/e, prof. Mykola and Vishnu for their valuable feedbacks and supervision.

This report is probably my final report as a student. I am sure the knowledge and experience I acquired during my bachelor's and master's will be of great use to me and the society. This report took a lot of effort, and I hope you will like this piece of work.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

-Alan Turing

Mayank Padhi

Contents

Chapter 1

Introduction

The production printers use inkjet technology that takes some time to dry for a sheet after a print. Due to the drying process, the color changes. Hence, the output may not be the same as it was intended to be. Hence, a prediction model is required such that it can predict the color after it has dried (aka *stabilized*). So, the output of the printer would be such that when the color has dried, we get the intended colors. This is particularly important while printing photographic images. The printer being used for the collection of data is Canon VarioPrint iX 3200 (hitherto called *the printer*) available at the Customer Experience Center, CPP Venlo.The printer uses the CMYK (A) ink. Hence, the input for a color is the target CMYK value. All the colors are printed using a combination of CMYK colors. There are a number of factors that influence the change of color. Horvath et al. [25] have provided a list of such factors.

- dryers in the ink,
- viscosity of the ink,
- quantity and quality of the lubricant used during printing,
- air temperature,
- ambient air humidity,
- paper surface,
- paper volume and aA value,
- air flow

However, the previous studies in the organization suggest that the print conditions have effect on the drying. The print conditions include the temperature, roll drum for steam treatment, color grip, ink amount, and print speed. These settings can be adjusted depending on the requirement. However, all of these conditions remained constant during the data collection. Furthermore, the study by Horvath et al. [25] touched upon the broader effect of media characteristics. In this report, we attempt to study the effect of each of these characteristics and use it to make predictions. The media characteristics consist of the properties of the paper which include the whiteness, brightness, bulk, thickness etc. We discuss about the media characteristics in detail in the subsequent chapters. Our study is focused on understanding the effect of these media characteristics on dry back while treating the other factors as constants.

1.1 Research Problem

In this study, we try to find out the different factors that contribute to the degree of change in color upon drying. Generally speaking, the time taken by the sheet to dry varies from 24 hours to about 72 hours. As suggested by Horvath et al. [25], the drying time and the degree of dry back depends on the different types of surface of paper (i.e., treated/coated/uncoated). Our study will focus on the coated media type. While collecting the data for each media, it is important to make sure that the color does not change after the measurements (or after it has stabilized) are taken. Hence, the regression model should take these factors into account, and provide a reliable prediction for the new media introduced to the printer in the future. Furthermore, we discuss the factors that determine the dry back.

1.2 Contributions

The main technical contribution of this report is to assist the printer operators to make a more reliable color profiler for the media so that it provides the best results for new papers without having to wait for days to let the color dry. While making a profile, if someone does not want to wait for the color to dry and use wet measurements (or the Default Media Entries (DME)), this can lead to inaccurate calibration and eventually imperfect printing. There have been some studies in the past to understand the dry back phenomena. However, the results were limited to predicting the dry back of selected media for test charts with few colors. This report provides a foundation, and sets a benchmark on the dry back effect by different media types under the given conditions.

1.3 Outline

The remainder is described as follows; The problem formulation is described in Chapter 3. Chapter 2 covers the preliminary of the concepts and basic assumptions in the report such as regression along with the notation and definitions used throughout this report. The literature survey is discussed in Chapter 4, the experiments conducted prior to the data collection, and the model is discussed in Chapter 5, the results, and observations are discussed in Chapter 6, and Chapter 7 concludes the research while discussing about the limitations and the future possibilities. The extended results, terms, and definitions, and some experiments that are not directly a part of this study are included in the Appendix A.

Chapter 2

Preliminaries

In this chapter, several definitions, assumptions, and notations are introduced which will be used throughout this report. In Section 2.1, we briefly discuss the concepts, notations, and assumptions about the dry back, and the collected data. In Section 2.2, we introduce regression, and the basic concepts around it. We also discuss the candidate models that can be used for prediction.

2.1 Data and Dry Back

In general terms, a medium is a surface used for printing. In the context of this project, we use medium in place of page/paper. The inkjet printers use a liquid ink which takes certain amount of time to dry. The dry back is an effect when the density and/or gloss of the freshly printed ink film decreases upon drying [25]. It is generally related to an overly absorbent paper surface or a poor ink/paper combination [25]. The dry back causes the color of the ink to change. The magnitude of δE helps in quantifying this change of color. Hence, if the δE is large the change is high, and if the δE is small, the change is low. The δE is discussed in detail in Chapter 4.

These color changes are identified printing, and scanning a test chart with a spectrophotometer. The data is collected at 3 points in time i.e., just after the print, after 24 hours, and after 96 hours (or 4 days). The data collected at these points in time are referred to as Time 00, Time 24, and Time 96 respectively. The Xrite ISIS spectrophotometer device was used to measure the $L*$, $a*$, and $b*$ values.

2.1.1 Test Chart

The test chart contains 760 color patches with 365 unique colors. Each color appears at least twice in the test chart. A study performed at CPP suggested that the location of the color patches affects the color measurements. In other words, the measurement of the $L*$, $a*$, $b*$ values of a color varies depending upon the location, if is located near the left or right edge or somewhere around the center. The inconsistency is because of the printing process, it also depends on the adjustment of the printer and the media itself. Hence, the patches have been placed in the test chart to take the effect of location into account. The Figure G.1 shows the test chart used in the research. The tables in Appendix G show the details of colors along with their CMYK values used in each patch in the test chart.

2.1.2 Concepts and Standards of Color

There are various aspects of color that are useful while defining a color. There are color spaces such as the $L * a * b*$ and XYZ color space that define a color. Moreover, the perception of color on a surface also depends on its illuminant (i.e. the type of light). Furthermore, we discuss about the concept of δE which is one of the key metrics used to evaluate the regression models.

$L*$ $a*$ $b*$ Color Space

The $L*$ $a*$ $b*$ color space (or CIELAB) was defined by the International Commission on Illumination (or CIE) in 1976. The $L*$ $a*$ $b*$ values signify the perceptual lightness, red-green, and blue-yellow respectively. As shown in Figure 2.1, the X-axis (or $a*$) goes from green to red, and the Y-axis (or $b*$) goes from blue to yellow. The values of $L*$ $a*$ $b*$ varies from $[0, 100]$, $[127, 128]$, and $[127, 128]$ respectively. This color space covers the entire range of human color perception.

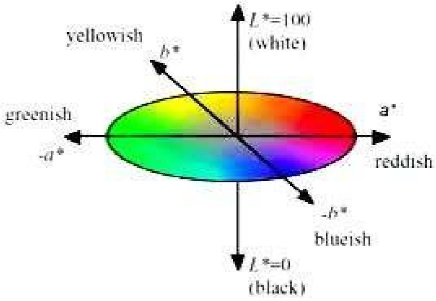


Figure 2.1: Representation of $L*$ $a*$ $b*$ color space[4]

As shown in Figure 2.1, the X-axis (or $a*$) goes from green to red, and the Y-axis (or $b*$) goes from blue to yellow.

Quantifying the Color-Difference: δE

Once, the colors have been quantified (with the $L*$ $a*$ $b*$ color space), the change in color can also be quantified just by finding the difference. Hence, δE is just the difference of the visual perception of two colors. However, the formula to compute δE has been changing over the years (CIE76, and CIE94 [44]). The version used in the report is the latest version introduced in 2000 by Sharma et al. [53]. The value of δE typically varies from 0 to 100. The significance of the values of δE is defined in Table 2.1.

δE	**Perception**
≤ 1.0	Not perceptible to human eye.
$1 - 2$	Perceptible through close observation.
$2 - 10$	Perceptible at a glance.
$11 - 49$	Colors are more similar than opposite
100	Colors are exactly opposite

Table 2.1: δE values with Perception.

The $L*$ $a*$, and $b*$ values are present in both the inputs (or X), and the outputs (or y) The difference is that the values in inputs are from Time 00, and the values in y are at Time 96 for a given color patch. Ideally the predictions should be same as the values Time 96 (or $y_{test} = y_{pred}$). However, we know that is not possible in practice. Hence, we try to minimize the difference between y_{test}, and y_{pred}. We quantify this difference by computing the δE between the y_{test}, and y_{pred}. Hence, this is the target δE we are trying to minimize with our regression model. Throughout this report (unless stated otherwise) the δE always refers to the δE between the y_{test}, and y_{pred}.

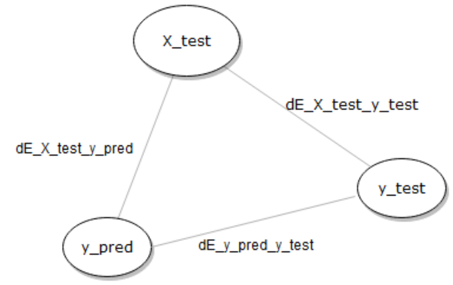


Figure 2.2: Pictorial representation of δE between different sets of $L*$ $a*$, and $b*$

In Figure 2.2, it is interesting to note that the $\delta E_{X\_test\_y\_test}$ signifies the actual dry back. Furthermore, we want the regression model to generate results (i.e., $L*$ $a*$, and $b*$) such that it is close to $\delta E_{X\_test\_y\_pred}$.

The definition of the modern δE was given by Sharma et al. [53]. The details of each step can be found in their work. However, a concise sequence of steps for a given pair of colors (i.e. $L^*_1 a^*_1 b^*_1$ and $L^*_2 a^*_2 b^*_2$) and weighting factors (i.e. k_L, k_C, and k_H) to find the δE is as follows [53].

- Calculate $C^{‘}_i$, and $h^{‘}_i$

$$\begin{aligned}
C^*_{i,ab} =& \sqrt{(a^*_i)^2 + (b^*_i)^2}, i = 1,2 && (2.1)\\
\overline{C}^*_{ab} =& \frac{\overline{C}^*_{1,ab} + \overline{C}^*_{2,ab}}{2} && (2.2)\\
G =& 0.5(1 - \sqrt{\frac{\overline{C*}^7_{ab}}{\overline{C*}^7_{ab} + 25^7}}) && (2.3)\\
a^{‘}_i =& (1+G)a^*_i, i = 1,2 && (2.4)\\
C^{‘}_i =& \sqrt{(a^{‘}_i)^2 + (b^*_i)^2}, i = 1,2 && (2.5)\\
h^{‘}_i =& \begin{cases} 0, & b^*_i = a^{‘}_i = 0 \\ tan^{-1}(b^*_i, a^{‘}_i) & otherwise \end{cases} i = 1,2 && (2.6)\\
& && (2.7)
\end{aligned}$$

- Calculate $\Delta L^{‘}$, $\Delta C^{‘}$, and $\Delta H^{‘}$

$$\begin{aligned}
\Delta L^{‘} =& L^*_2 - L^*_1 && (2.8)\\
\Delta C^{‘} =& C^{‘}_2 - C^{‘}_1 && (2.9)\\
\Delta h^{‘} =& \begin{cases} 0, & C^{‘}_1 C^{‘}_2 = 0 \\ h^{‘}_2 - h^{‘}_1, & C^{‘}_1 C^{‘}_2 \neq 0; |h^{‘}_2 - h^{‘}_1| \leq 180^o \\ h^{‘}_2 - h^{‘}_1 - 360, & C^{‘}_1 C^{‘}_2 \neq 0; (h^{‘}_2 - h^{‘}_1) > 180^o \\ h^{‘}_2 - h^{‘}_1 + 360, & C^{‘}_1 C^{‘}_2 \neq 0; (h^{‘}_2 - h^{‘}_1) < -180^o \end{cases} && (2.10)\\
\Delta H^{‘} =& 2\sqrt{C^{‘}_1 C^{‘}_2} sin(\frac{\Delta h^{‘}}{2}) && (2.11)\\
& && (2.12)
\end{aligned}$$

- Calculate δE_{00} (or just δE)

$$\overline{L}^{‘} = (L_1^* + L_2^*)/2 \tag{2.13}$$

$$\overline{C}^{‘} = (C_1^{‘} - C_2^{‘})/2 \tag{2.14}$$

$$\overline{h}^{‘} = \begin{cases} \frac{h_2^{‘}+h_1^{‘}}{2}, & C_1^{‘}C_2^{‘} \neq 0; |h_2^{‘} - h_1^{‘}| \leq 180^o \\ \frac{h_2^{‘}+h_1^{‘}+360}{2}, & C_1^{‘}C_2^{‘} \neq 0; (h_2^{‘} + h_1^{‘}) < 360^o; |h_2^{‘} - h_1^{‘}| > 180^o \\ \frac{h_2^{‘}+h_1^{‘}-360}{2}, & C_1^{‘}C_2^{‘} \neq 0; (h_2^{‘} + h_1^{‘}) \geq 360^o; |h_2^{‘} - h_1^{‘}| > 180^o \\ h_2^{‘} + h_1^{‘}, & C_1^{‘}C_2^{‘} \neq 0 \end{cases} \tag{2.15}$$

$$\tag{2.16}$$

$$T = 1 - 0.17cos(\overline{h}^{‘} - 30^o)) + 0.24cos(2\overline{h}^{‘}) + 0.32cos(3\overline{h}^{‘} + 6^o) - 0.20cos(4\overline{h}^{‘} - 63^o) \tag{2.17}$$

$$\Delta\theta = 30exp\left\{ - \left[\frac{\overline{h}^{‘} - 275^o}{25} \right] \right\} \tag{2.18}$$

$$R_C = 2\sqrt{\frac{\overline{C}^{‘7}}{\overline{C}^{‘7} + 25^7}} \tag{2.19}$$

$$S_L = 1 + \frac{0.015(\overline{L}^{‘} - 50)^2}{\sqrt{20 + (\overline{L}^{‘} - 50)^2}} \tag{2.20}$$

$$S_C = 1 + 0.045\overline{C}^{‘} \tag{2.21}$$

$$S_H = 1 + 0.015\overline{C}^{‘}T \tag{2.22}$$

$$R_T = - sin(2\Delta\theta)R_C \tag{2.23}$$

$$\delta E_{00}^{12} = \delta E_{00}(L_1^*, a_1^*, b_1^*, L_2^*, a_2^*, b_2^*) \tag{2.24}$$

$$= \sqrt{\left(\frac{L^{‘}}{k_L S_L}\right)^2 + \left(\frac{C^{‘}}{k_C S_C}\right)^2 + \left(\frac{H^{‘}}{k_H S_H}\right)^2 + R_T\left(\frac{C^{‘}}{k_C S_C}\right)^2\left(\frac{H^{‘}}{k_H S_H}\right)^2} \tag{2.25}$$

$$\tag{2.26}$$

For most of the scenarios, $k_L = k_C = k_H = 1$.

2.1.3 Selection of Media

The media selection is a vital step as there are a lot of manufacturers (based on different parts of the world) that provide a range of offset coated media. Hence, it is not feasible to include each of these types in the data set. However, there is chance to get restricted to a small range of δE if the dataset is not carefully chosen. Since, the drying effect of each medium has not been studied before, the selection of the media was a challenge. Discussions with domain, and market experts helped us to determine the prominent manufacturers, and their range of products that are available at the CPP warehouses that can show substantial drying effect. Broadly, there are the following types of papers [31] (Table 2.2).

Media Type	Coated/Uncoated
Offset coated	WFC
Inkjet coated	Digital/Special
Speciality Board	WFC
Speciality Embossed	WFU
Speciality NCR	Reduced Ink Volume
Inkjet treated Uncoated	WFU
Uncoated	WFU

Table 2.2: Types of papers/media

A study by conducted in the past suggested that uncoated media have relatively small δE. It is in the order of $\delta E_{95} \in [0.3, 0.7]$. Since, the change in color is not significant, a prediction model would not be necessary for the WFU types. The domain experts suggested that the offset coated type is most commonly used in the industry. Hence, this report is focused only on the offset coated type.

There are 3 broad types of coating materials used by different paper suppliers. These are *gloss*, *silk*, and *matt*. These variations have been covered by the dataset. Table H.1 provides the list of media used in this study.

We conducted a study to understand the effect of weight on δE (discussed in Section 5.3), we used the 6 different weight categories of the Magno Gloss by Sappi (EU). The choice of the medium was based on availability of a wide range of weights for a medium. The weights categories are mentioned in the Table 2.3.

115 gsm	135 gsm	150 gsm	250 gsm	300 gsm	350 gsm

Table 2.3: List of weights used to study the impact of weight

2.2 Regression

Regression is a statistical technique to study the relationships between variables [2]. These relationships help in making predictions of certain quantities based on the values of other quantities. It is used when we want to predict a continuous dependent variable using a number of independent variables. Regression analysis finds its use in time series forecasting, modeling, finding the relation between the variables and predict continuous values. There are various types of regression techniques, and the choice can be made based on the data set, and the end goal. The most basic regression technique is Linear Regression which uses the linear regression line to best fit the linear relationship. There are certain assumptions that the data has to satisfy for it to be effectively used for linear regression. These include linear relationship between dependent variables and features, normality of residuals, multicollinearity, no autocorrelation in features and homoscedasticity of error terms. The tests on the data suggest that most of the features do not have linear relationship with the dependent variables. Moreover, the Anderson-Darling test [3] on the residuals suggests that the residuals are not normal.

The data set of this project is heterogeneous and some features (such as CMYK) do not follow any well-defined distribution. Hence, there is a need to explore the techniques that do not have such underlying assumptions and constraints. The approach for various regression algorithms differs greatly and hence the accuracy of these models also varies as well (for a given problem) for each of these. In Chapter 4, we discuss the regression technique we use in greater detail. Further, in Chapter 5, we discuss the design choices for each of the regression algorithm in the context of our problem.

2.3 Summary

In this chapter, we discussed the concept of dry back, δE, and the test chart. This laid the ground for evaluation of the regression models in the context of this problem. We also discussed the different types of media, the motivation behind the choices, and their importance in the study. Later, we discussed the basic concept of regression, and discussed some commonly used regression techniques. In Chapter 3, we describe the problem, and define scope of this project, as well as discuss the approach to answer the research questions.

Chapter 3

Problem Description

In this chapter, we discuss the context of the problem in Section 3.1. The approach to answer the research questions is described in Section 3.2.

3.1 Problem Context

The color profiles are available in the printer's software which calibrate the settings for each medium. However, when a new medium is introduced, the operator has to either take a print of the test chart, wait for the color to dry and then make a color profile or just use a default media entry for the coating. While the first approach gives better quality outputs, it takes time to perform. Whereas if the second approach is taken, it is instantaneous as it does not even need a sample printout with the medium but there is a compromise with the quality. Hence, with the prediction model, we intend to save the time of the printer operator while maintaining the quality.

This project is a proof of concept to determine if it is possible to predict the dry back of a media based on its properties and make an accurate color profile. Finding the factors that affect the dry back is important as these results could be a good starting point if a coated media needs to be manufactured with focus to address the problem of dry back. The results from this study can also be used to identify the need to install stand-alone sensors to collect data on an unknown medium introduced to the printer since the data required to accurately predict the dry back may not be available in advance.

In Chapter 2, we discussed that the color changes when the ink is absorbed by the surface of the medium. Among all the factors, this project primarily focuses on the effect of media on dry back (keeping the other factors as constants). The problem is to understand:

What are the factors that impact the drying behavior? How can this drying effect be predicted under different printing and media conditions?

There are measures that are typically used to evaluate the performance of the regression algorithm such as the *mean squared error* (or MSE), *mean absolute error* (or MAE), r2 score and accuracy score. However, in the context of our research problem, we add another measure to evaluate the quality of the predictions i.e., δE. The quantity has been described in Chapter 2. Since, the δE is computed for each color patch, we use the aggregations of δE i.e., the average δE (or δE_{avg}), the 95 percentile of δE (or δE_{95}) and maximum δE (or δE_{max}). If the δE of most of the color patches is less than 1, we can say that the change of color cannot be perceived by humans. However, considering the errors in printing and measurement, the goal of our prediction model is to have a $\delta E_{95} < 0.5$.

3.2 Research Framework

The research problem can be divided into 2 parts:

- Finding the factors that affect the drying behavior.
- Predicting the change of color based on these factors.

In this study, we focus only on the factors around media. Any medium has a number of characteristics that play a key role in printing. These include its media characteristics such as the weight, thickness, brightness, whiteness, opacity, gloss, bulk etc. These characteristics are described in detail in the subsequent chapters. All these characteristics might not have equal impact on the dry back phenomena. The domain knowledge can provide a route to solve this problem. The domain experts suggest that the dry back is a surface phenomenon and depends on the size of the pores at the surface of medium (along with other factors of ink which are constant to us). Moreover the pores in the surface affect the gloss, opacity and brightness. Furthermore, the effect of dry back (or δE) can also be studied by finding the correlation of all the inputs with the δE (computed between Time 00 and Time 96). In fact, this analysis can also be performed while working on the second sub-problem. Once a *good* regression model is implemented and trained, the feature importance score of each of the inputs (features) would provide information about the features that contributed more for the accurate prediction.

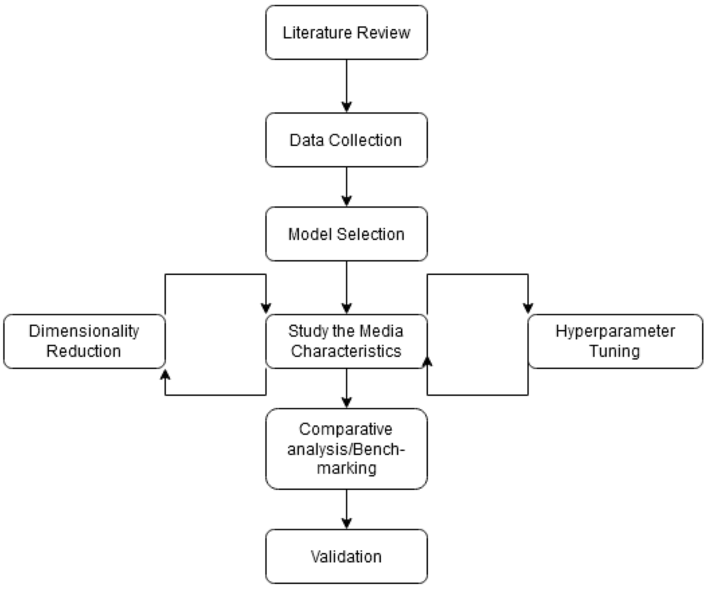


Figure 3.1: Flow Chart of the steps for the project

In order to solve the second problem, we need to fix an initial data set with a broad set of features. Then some candidate regression algorithms can be selected based on the composition of the data set. By composition, we mean the type of features i.e., numerical or categorical, the scale of data in each of the numerical features, the distribution of data in each of the numerical features, the number of outliers in each of the numerical features etc. Once the candidate regression algorithms have been implemented, they can be tuned for the best set of hyperparameters. Those models need to be tested and validated before their final results can be compared with each other to select the best candidate. Similar problems have been solved in the past in CPP by implementing some regression model.

3.3 Summary

In this chapter, we defined the problem statement and the scope of this project. We also discussed the approach to solve the problem and the potential challenges. In Chapter 4, we dive deeper into the relevant concepts used in this report such as δE, $L*$, $a*$, $b*$ color space. We also discuss some of the previous models used to predict these quantities. Furthermore, we also discuss the background of the regression techniques we use.

Chapter 4

Literature Survey

In this chapter, we discuss the concepts used in this report. In Section 4.1, we discuss the media characteristics that are available in the dataset. In Section 4.2, we study, and evaluate the regression models that have been used by CPP, and ISO in the past. In Section 4.3, we propose some regression models for implementation based on the collected data.

4.1 Media Characteristics

Media characteristics are the properties of the paper that it exhibits due to its composition. The suppliers of these media keep these information public to assist the printing companies, and customers regulate the printer settings based on this data for best quality outputs. Each supplier has its own set of procedures, and techniques which it uses to manufacture the paper. These techniques are generally confidential but some of the key factors that determine the media characteristics are publicly available as media specifications in the suppliers' website. There are several media characteristics, some of them are listed as follows:

- **Thickness:** It is defined as the thickness of a sheet of paper under specific conditions. Generally, the thickness is measured in *nm* or thousandths of an inch or *mils*. Most of the suppliers use unit, and method of measurement in accordance with ISO 534 [30].
- **Opacity:** Opacity is the characteristic of medium to block the transmission of light. It is measured, and expressed as a percentage of the light that cannot pass through the medium. Hence, 95% opacity means that 95% of the light cannot pass through the medium. Most of the suppliers measure, and communicate the opacity in accordance with ISO 2471 [28].
- **Whiteness:** It is measured in CIE Whiteness C/2-degree (indoor illumination conditions) or CIE D65/10 (outdoor illumination conditions). It is communicated on the basis of measurements made in accordance with ISO 11475 [27].
- **Gloss:** Gloss is the shininess or glare reflected from surface of a medium. It is the reflection of light when the angle of incidence, and reflection is kept at a standard angle from the surface, as compared to a polished plate of black glass. The angle of incidence varies depending upon the ISO standard being used. For papers, and boards, ISO 8254-1 [29] has 75 degrees with a converging beam, ISO 8254-2 [32] has 75 degrees with parallel beam, and ISO 8254-3 [33] has 20 degrees with converging beam. It is expressed in percentage, and the medium may be termed as dull or glossy depending upon the measure of gloss.
- **Brightness:** It is the amount of light reflected from the surface of a medium, compared to light reflected from a block of Magnesium Oxide. The measurement is made with a specific range of wavelength of light, with the surface of the medium being illuminated at a 45-degree angle, and the reflection being measured at a 90-degree angle. The brightness

influences the printed contrast, and the amount of reflected illuminating light. It is measured, and communicated in accordance with ISO 2470-1 [34] or ISO 2470-2 [35] in percentage.

- **Bulk:** It is the measure of the density with respect to the weight of a sheet of paper. The unit used is pages per inch (PPI). Individual sheets do not necessarily add to the PPI as it may vary depending on how the sheets stack together. Most of the suppliers use unit, and method of measurement in accordance with ISO 534 [30].

- **Smoothness:** Also known as finish, smoothness is the texture of the surface of the medium. It is determined by measuring the flow of air along the surface of a medium under standardized loading, thermal, and pressure conditions. The faster the flow of air, lesser is the smoothness. It is communicated on the basis of measurements made in accordance with the ISO 5627 [26].

- **Fluorescence:** The fluorescence is the rays produced from the surface as a result of incident light of a shorter wavelength. It is communicated on the basis of measurements made in accordance with ISO 11475 [27].

Further details about these standards are beyond the scope of this report. Since, these media characteristics are provided by the suppliers, they do not provide the data about all the characteristics. Hence, using these characteristics in the model introduces a risk of bias.

4.2 Previous Approaches to the Problem

There have been some attempts in the past to predict the dry back. Some have used the spectral reflectance data as inputs, and outputs. Before using it in the model they remove the surface diffusion component from the dried, and wet data.

A similar project was undertaken at CPP in the past to predict the $L*$, $a*$, and $b*$ values. They used three different algorithms viz. Multi-layer Perceptron (MLP) model, LASSO regression, and Ridge regression. The LASSO, and Ridge regression yielded a MSE of 0.24, and 0.25 respectively. Moreover, the input dataset varied by each model.

Their MLP model has 3 different variations, each with a different number of inputs. The first model has 7 inputs, C, M, Y, K, and $L*$, $a*$, $b*$ at Time 00, and the outputs has the $L*$, $a*$, $b*$ values when the color has stabilized. The second MLP model contained both coated, and uncoated media. Hence, a binary input was included in the model to signify if the model was coated/uncoated. The third MLP model added 3 more inputs i.e. the $L*$, $a*$, $b*$ values after 5-10 mins of the first measurements. The hidden layer in all these models has 13 neurons. In all the three MLP models, output variables remained the same i.e. $L*$, $a*$, $b*$ after the color had stabilized. The time interval between the measurements varied by each medium. However, this is less likely to affect the results as it was ensured that the color had stabilized. In fact, the data from this study helped us restrict the dataset to coated media as coated media showed significant change in color when compared to uncoated media. Moreover, this study also helped us set the duration between the 2 measurements to ensure that the color of all the media had stabilized.

The dataset used for training, and testing was limited to Magno Gloss, Magno Matt, Magno Silk, Mitsubishi, Soporset. When the training, and testing dataset contained a single medium, the 95 percentile of δE varied from 0.88 to 1.49. However, the model thus trained cannot be used for predicting other media. The results suggest that the regression models were not scalable, and required more data, and complexity. This study laid the groundwork for the project described in this report.

The previous approaches also made use of LASSO, and Ridge regression algorithms. The 7 features of the data (i.e. C, M, Y, K, and $L*$, $a*$, $b*$ at Time 00) which were scaled down using Robust Scaling before they were used for training. The media covered was the same as the aforementioned list. Almost all of the hyperparameters of both the algorithms remained retained the default values except the regularization factor *alpha* which was 0.0001, and 0.01 respectively.

This resulted in the MSE_{test} of these algorithms to be 0.280, and 0.250 respectively. However, these models lacked the generalizability over different weights, and types of media.

In these approaches, we observe that there is no feature in the training data that differentiates between two (or more) media (and its properties) with an exception of one model that used a binary input to differentiate between the coated and uncoated media. Since we know from these experiments that the degree of drying varies by media, we need to use this information (and more) to effectively use in the model so that it makes more accurate and informed predictions. Hence, we study the properties of each of these media and make use of them for prediction.

4.3 Regression Techniques

The studies have shown that the extent of drying varies for each media. However, it is not known exactly what features determine this variation across different media. The results from the previous studies in the organization did not take the media into account. This suggests that more data (more media and more features for each medium) should be collected to get a clearer picture of the relationships. After collecting the data, we start with considering the MLP model just like the one used in the past with an increase the number of inputs. Furthermore, the model can be made more complex by adding more hidden layers, and including embedding layers for categorical inputs.

The data set contains a variety of features such as categorical features (such as supplier), numerical features that do not follow a standard distribution ($CMYK$), features at different scales etc. This poses a limitation to the regression techniques that can be used. Hence, we need some regression techniques that can provide robust results with such data and do not have many constraints. Moreover, we also need regression models that are able provide interpretable solutions because we also study the role media characteristics play in determining the dry back.

Introduced in 1963 by Morgan et al. [45], the tree-based regression techniques provide simple and efficient solutions without any condition on the data. Moreover, the regression trees are very flexible regression methods as they do not need any major preprocessing (such as scaling/normalization) for optimal performance. Furthermore, the obtained models are relatively more interpretable. The ensemble regression trees can be broadly classified into 2 types: bagging, and boosting. Both the techniques are known to generate interesting yet useful results. Additionally, the support vector machines are also able to model complex data, and are tolerant to outliers. The following algorithms are useful to the project, and can be explored further:

4.3.1 Decision Tree Regression

In decision tree modeling, a tree represents a division of data created by applying a series of trivial rules. These trees are made from these set of rules which can be used for prediction through the repetitive process of splitting. It is one of the most commonly used methods for classification, and regression problems [58]. The most effective splitting strategy is to use the entropy of split. An advantage of the decision tree is that it produces a model which can represent interpretable rules. In other words, it can provide useful information on the importance of each feature for prediction [58]. However, decision tree induction generally does not perform as good as neural networks for nonlinear data, and it is vulnerable to noisy data [13]. But the tree-based algorithms are more effective when the data contains categorical values.

4.3.2 Ensemble Learning

Before we discuss about the other tree-based regression algorithms, let us discuss the concept of ensemble learning. One of the earliest researches on ensemble learning was by Dasarath et al. [14] for solving classification problems. The ensemble learning is the process by which a number of regression models are generated, and their combined result is used to make predictions. It is used to improve the prediction of a model, and to reduce the chances of selection of a *bad* model. There are other applications of ensemble learning, such as classification, selecting optimal features, data fusion, incremental learning etc. There is no guarantee that a combination of multiple regressors would always perform better than the best model in the ensemble. Moreover, an improvement on the ensembles average performance cannot be guaranteed in most of the cases [20]. Hence, combining regressors may not necessarily provide the best performance in the ensemble, but it would reduce the overall risk of making a bad selection of model. An advantage of the ensemble learning is that unlike the MLP, it needs few tuning parameters, and in most cases default configuration yields satisfactory performance. The results for each of these algorithms is discussed in the Section 6.

There are broadly 3 types of ensemble learning viz. Bagging, Boosting, and Stacking. In this report, we cover 2 out of these 3 types of ensemble learning. They are described as follows.

Bagging

It is one of the earliest, and the simplest ensemble learning technique with a good performance [6]. Diverse regressors in bagging are obtained by bootstrapping the replicas of the training data. Individual regressors are then combined by taking a simple aggregation of their decisions. Since the training datasets may overlap (due to bootstrapping), additional measures can also be used to increase diversity, such as using a subset of the training data for training each regressor or using relatively weak regressors (such as stumps).

- **Random Forest Regression**

  Random forest is a most popular bagging technique. As the name suggests, it is a *forest* of decision trees. The trees in random forests run in parallel. Hence, there are no interactions between the trees while building the trees. There are 2 unique properties of the random forest regression. There is a limitation to the number of features that can be split on at each node. Hence, the ensemble model is able to make fair use of all potentially predictive features. An informed choice is made during the splitting in each node. This ensures the best decision for a split in each node. The SciKit Learn [47] library provides 2 such criteria for splitting. The default splitting criteria is the Gini Impurity. Gini is the probability of correctly predicting a randomly chosen element if it was randomly predicted according to the distribution of independent variables in the node [36]. The Gini criterion for the parent node is always higher than its child-nodes [36]. Entropy is another criteria for splitting of the nodes. Entropy is also used for calculating the purity of a node. Lower is the entropy, higher is the purity of the node.

- **Extra Trees Regression**

  Extremely randomized trees (or extra trees) [21] algorithm is very similar to the random forest. In principle, it is an extension of random forest algorithm, and unlike the random forest it is less likely to overfit [21]. Extra trees employ the same principle of building lot of decision trees as random forest, and use a random subset of features to train each base estimator [37]. However, unlike the random forest, the extra tree randomly selects the best feature, and its corresponding value for splitting the node [37]. The extra tree uses the whole training dataset to train each regression tree unlike the random forest which uses bootstrapping during training [1].

  These differences result in the reduction of bias, and variance for extra tree. Using the whole original sample instead of bootstrapping reduces bias, and choosing random split point for each node reduces the variance [21]. It is also interesting to note that the time taken by the extra tree regression algorithm to evaluate is lower than that of the random forest. It can be attributed to the fact that there no bootstrapping in extra trees, and for each node time is saved by not computing the best strategy to split.

Boosting

Boosting was introduced in around 1990 by Schapire et al. [51]. It is another ensemble technique to create a collection of predictors. In this technique, the learning happens sequentially with early learners fitting simple models to the data, and then analyzing data for errors. Hence, the model learns sequentially in an adaptive way (depends on the previous steps), and then combines the results using a deterministic strategy [16]. The working mechanism is similar to the bagging methods: we construct a forest of models that are aggregated to obtain a strong learner that performs better. Unlike the bagging method that aims at reducing variance, the boosting method fits the multiple weak learners sequentially in an adaptive way. Each model in the sequence is fitted giving more importance to observations in the data set that had bad results by the previous models in the sequence [16] [41]. The base models mainly focus at reducing bias; hence we begin with shallow decision trees with high bias (and low variance).

A drawback of this algorithm is the computation time. Unlike bagging, the training of these models cannot be done in parallel. As a result it could be an expensive operation to fit these complex models sequentially. We need to find out how we evaluate the models in the sequence. In other words, we need to find what information from the previous model do we need to consideration while fitting the current model. In this report, we discuss the three most commonly used variants i.e. Adaptive Boosting (also known as AdaBoost), Gradient Boosting, and XG Boosting regression. The Adaptive Boosting updates the weights attached to each observation whereas Gradient Boosting updates the value of these observations.

- **Adaboost Regression**

  Introduced by Freund et al. [19], the AdaBoost (Adaptive Boost) algorithm involves very shallow (generally just one-level) decision trees as weak learners (also known as stumps) that are added sequentially to the ensemble. A weak learner is a tree that is very shallow, although it does provide some insight on the data set. Each subsequent weak learner attempts to correct the predictions made by the model before it in the sequence. This is achieved by weighing the training dataset to put more focus on the same training examples on which prior models made prediction errors. The AdaBoost algorithm puts more weight on difficult features (to predict), and less on those already handled well.

- **Gradient Boost Regression**

  The Gradient Boosting algorithm is very similar to the Adaptive Boosting algorithm. Just like the other boosting techniques, it also trains many models in a gradual, additive, and sequential way. The major difference between AdaBoost, and Gradient Boost regression algorithms is how the two algorithms identify the shortcomings of weak learners (i.e. decision trees or stumps). While the AdaBoost identifies the shortcomings by using high weight data points, Gradient Boost performs the same by using gradients in the loss function. The loss function is a function indicating the models performance at fitting the training data.

- **Extreme Gradient Boost Regression**

  Introduced in 2016 by Chen et al. [10], the eXtreme Gradient Boosting (also known as XGBoost) regression technique is similar to its predecessor, the Gradient Boosting regression. In fact, the prime difference between XGBoost and Gradient Boosting method is that XGBoost uses more accurate approximations to find the best tree model (which are relatively expensive). It computes the second partial derivatives of the loss function to get more information about the direction of gradients, and how to get to the minimum of our loss function [11] [10]. This step is similar to the Newton-Raphson's method to find approximate the root of a function. While the gradient boosting uses the loss function of the model (e.g. Decision Tree) as a measure for minimizing the error of the overall model, XGBoost always uses the 2nd order derivative as an approximation. As a result, the training is relatively faster, and can also be parallelized. XGBoost also includes a variety of regularization techniques (such as L1, and L2) that reduce overfitting, and improve overall performance. You can select the regularization technique by setting the hyperparameters of the XGBoost algorithm [10].

4.3.3 Support Vector Regression

The concept of support vector was first introduced at AT&T Bell Laboratories by Vapnik et al. [5] [23]. The support vector machines (SVM) is a popular classification technique that classifies the data by making a hyperplane between the two (or more) classes. However, similar approach has also been used for regression. The SVR is somewhat similar to the linear regression techniques. The main objective is to minimize the squared errors. For example, in Ordinary Least Squares (OLS) the objective function (for one predictor) is

$$min \sum_{i=1}^{n} (y_i - w_i x_i)^2 \tag{4.1}$$

where y_i is the target variable, w_i is the weight, and x_i is the independent variable (or feature).

SVR has the flexibility to have a range for acceptable error in the regression model, and will find an optimal hyperplane to fit the data. Hence, unlike OLS the objective of SVR is to minimize the L2-norm of the coefficient vector (and not the squared error) [54]. In fact, the absolute error term is constrained to be less than or equal to a specified margin (Equation 4.3).

Minimize:

$$min \frac{1}{2} \|w\|^2 \tag{4.2}$$

Constraint:

$$|y_i - w_i x_i| \leq \epsilon, \tag{4.3}$$

where ϵ is the maximum error.

Since, we can define the maximum error, there are always chances that the data will lie outside the margins $[w_i x_i - \epsilon,\, w_i x_i + \epsilon]$. These deviations from the margin can be denoted as ξ[54]. Hence, the Equations 4.2, and 4.3 become

Minimize:

$$min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} |\xi| \tag{4.4}$$

Constraint:

$$|y_i - w_i x_i| \leq \epsilon + |\xi| \tag{4.5}$$

where C is the regularization parameter.

4.4 Summary

In this chapter, we discussed the media characteristics, some past implementations, and potential regression algorithms to explore. In Chapter 5, we dive into the implementation of each step beginning with the data collection, data preprocessing, and implementation of each of the aforementioned algorithms.

Chapter 5

Implementation

In this chapter, we discuss the entire implementation phase of this project. We begin with configuring the spectrophotometer to collect the data (Section 5.1), and obtaining the accuracy scores, and using it to collect the data (Section 5.2). In the preliminaries, we had discussed the motivation, and the choice of media we covered in the dataset. We then find the stabilization time of each of the media, and set a standard interval for stabilization (Section 5.5). Then, we discuss briefly about the data on media characteristics, and the limitation around it (Section 5.3.2). We begin the implementation with studying the data, and preprocessing it (Section 5.6). This step varied depending upon the regression algorithm, however, the broader steps remained the same. Later, we dive into dimensionality reduction, and feature selection (Section 5.7). Finally we discuss the theoretical concept, and the implementation aspect of each of the regression techniques we introduced in Section 5.8.

5.1 Measurement Accuracy Test

Before the $L*$ $a*$, and $b*$ values are collected using the spectrophotometer, it needs to be tested for accuracy as the level of accuracy of any device deteriorates over a long period of time. This would help us set an upper limit to the accuracy of the model. This is because we cannot expect the model to make the predictions to the accuracy that cannot be measured. To conduct this test, the printed test chart is measured multiple times (seven times in our case) after the color has stabilized (i.e., at least after a 3-4 days of printing). The medium or media settings do not have any impact as the measurements are taken repeatedly for the same setting. For our study we used the Magno Satin (115 gsm) by Sappi. The $L*$, $a*$, $b*$ values were measured after 120 hours. Hence, it can be assumed that the color had stabilized during the measurement.

Ideally, the $L*$, $a*$, $b*$ values are supposed to be the same, as all the conditions are same for each measurement. Hence, the difference between any two measurements (i.e., the δE) helps us determine the measurement error in the data. This helps in setting up a limit for the accuracy of the prediction model. In other words, if the uncertainty in a measurement is x, the accuracy of prediction cannot be lower than x.

5.2 Data Collection

In this section, we describe the sequence of the steps taken to get the required data. It begins with selecting the test chart that is used for printing. It is important to have a wide range of colors (technically, the combination of $CMYK$ values) as experts have suggested that different colors have varying an impact on the δE. Then the media to be used in the research is determined. The colors are obtained as printed outputs generated by Canon Varioprint ix 3200. These outputs are a grid of different color patches. These are then inserted in a spectrophotometer, Xrite i1 Isis. The output contains the $CYMK$, and the $L*$, $a*$, and $b*$ values of each color patch.

However, there are certain aspects of a medium that would determine the variety we need to have in our data set. The impact of media grammage (or weight) on the dry back has not been widely studied. If the impact of weight is very significant, we may need to include a wide range of weight categories in each medium. If the weight does not a significant impact, we can limit the variation of weight in each of the medium. Hence, in the next section we describe the setup the experiment to study the effect of weight.

5.3 Effect of Weight on Dry Back

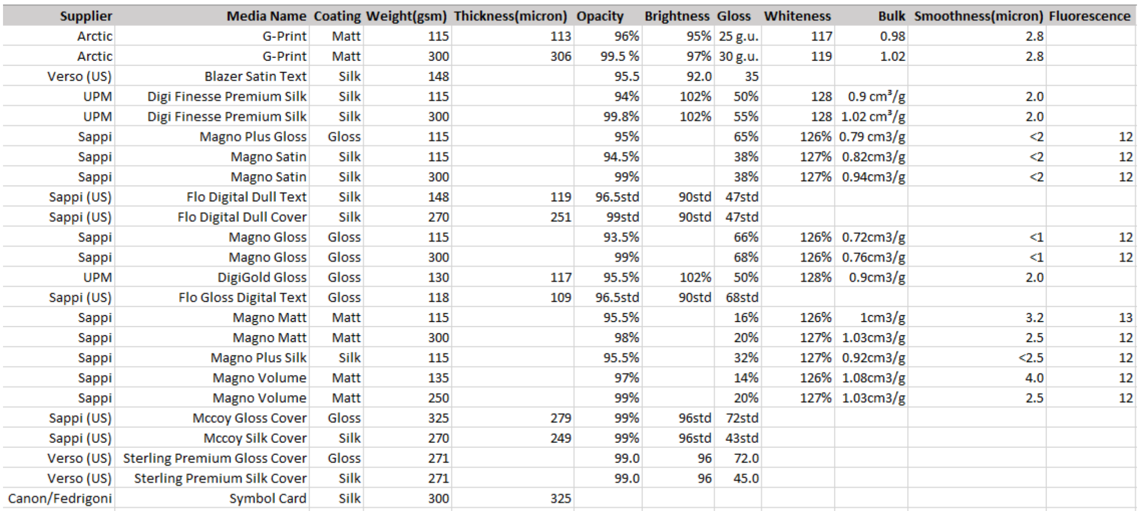
The study was conducted on different weights of Magno Gloss. In this experiment, we employ statistical approaches along with the domain knowledge was applied to understand the effect of weight on δE. Hence, the research questions are: *Is there any trend in δE by weights? If yes, how significant?*

5.3.1 Setup

Data was collected from a single medium type with 6 different weights. All the other print settings were the same to maintain consistency in the results. The TAC limit was set to $16.8ml/m^2$ opposed to the default $12ml/m^2$ to be consistent with the media settings for the original data. The $L*$, $a*$, $b*$ values were measured at Time 00, Time 24, and Time 96. However, Time 00, and Time 96 were used while computing the δE.

5.3.2 Media Characteristics

We employ the domain knowledge to evaluate the usefulness of the features. Hence, in this section we discuss the media characteristics. However, all characteristics are not provided by all the suppliers. The media characteristics are mentioned in Figure 5.1. Since data on opacity, and gloss were provided by almost all the suppliers, they have been considered to be included in all the regression models. The units of the quantities have been mentioned along with the number. Units have not been specified for some of the quantities as it was not provided by the supplier. The data on these media characteristics are not available in the organization at this moment.



Supplier	Media Name	Coating	Weight(gsm)	Thickness(micron)	Opacity	Brightness	Gloss	Whiteness	Bulk	Smoothness(micron)	Fluorescence
Arctic	G-Print	Matt	115	113	96%	95%	25 g.u.	117	0.98	2.8	
Arctic	G-Print	Matt	300	306	99.5 %	97%	30 g.u.	119	1.02	2.8	
Verso (US)	Blazer Satin Text	Silk	148		95.5	92.0	35				
UPM	Digi Finesse Premium Silk	Silk	115		94%	102%	50%	128	0.9 cm³/g	2.0	
UPM	Digi Finesse Premium Silk	Silk	300		99.8%	102%	55%	128	1.02 cm³/g	2.0	
Sappi	Magno Plus Gloss	Gloss	115		95%		65%	126%	0.79 cm3/g	<2	12
Sappi	Magno Satin	Silk	115		94.5%		38%	127%	0.82cm3/g	<2	12
Sappi	Magno Satin	Silk	300		99%		38%	127%	0.94cm3/g	<2	12
Sappi (US)	Flo Digital Dull Text	Silk	148	119	96.5std	90std	47std				
Sappi (US)	Flo Digital Dull Cover	Silk	270	251	99std	90std	47std				
Sappi	Magno Gloss	Gloss	115		93.5%		66%	126%	0.72cm3/g	<1	12
Sappi	Magno Gloss	Gloss	300		99%		68%	126%	0.76cm3/g	<1	12
UPM	DigiGold Gloss	Gloss	130	117	95.5%	102%	50%	128%	0.9cm3/g	2.0	
Sappi (US)	Flo Gloss Digital Text	Gloss	118	109	96.5std	90std	68std				
Sappi	Magno Matt	Matt	115		95.5%		16%	126%	1cm3/g	3.2	13
Sappi	Magno Matt	Matt	300		98%		20%	127%	1.03cm3/g	2.5	12
Sappi	Magno Plus Silk	Silk	115		95.5%		32%	127%	0.92cm3/g	<2.5	12
Sappi	Magno Volume	Matt	135		97%		14%	126%	1.08cm3/g	4.0	12
Sappi	Magno Volume	Matt	250		99%		20%	127%	1.03cm3/g	2.5	12
Sappi (US)	Mccoy Gloss Cover	Gloss	325	279	99%	96std	72std				
Sappi (US)	Mccoy Silk Cover	Silk	270	249	99%	96std	43std				
Verso (US)	Sterling Premium Gloss Cover	Gloss	271		99.0	96	72.0				
Verso (US)	Sterling Premium Silk Cover	Silk	271		99.0	96	45.0				
Canon/Fedrigoni	Symbol Card	Silk	300	325							

Figure 5.1: Media Characteristics

The percentage of missing data of the media characteristics varies from 37.5% to 62.5% Please refer to Table 5.1 for details. There are some techniques that can be used since there are missing values in the dataset. In this situation, the data is Missing At Random (MAR) [42]. The data is MAR if the missingness depend on other observed information in the dataset. In this case, the missingness of the medium characteristic depends on its supplier.

Thickness	**Opacity**	**Whiteness**	**Gloss**	**Brightness**	**Bulk**	**Smoothness**	**Fluorescence**
62.4%	4.2%	37.5%	4.2%	45.8%	37.5%	37.5%	58%

Table 5.1: Percentage of missing data on media characteristics

Bootstrap is a resampling method which allows estimation of the sampling distribution of almost any statistic using random sampling methods [52]. The bootstrap resampling is asymptotically valid, and effective irrespective of the sampling design or the imputation method [52]. However, since the percentage of missing data is very high (at least a third of data), these methods may not provide reliable data for prediction. The results from feature importance can potentially generate a need to install sensors to measure these characteristics in the upcoming printers such as *Mogami*.

However, while there are many factors (such as thickness and size of pores in the coating of the sheet) that are useful in determining the drying effect, this information are not available for all media. In fact, this information may or may not be available when a new media is introduced to the printer. Hence, the tests are motivated by the opinions of the domain experts, keeping in mind the scope, and limitations with the new media.

5.4 Experimental Setup

In this section, we discuss the spectrophotometer's configuration while collecting the data. To maintain consistency, we will keep these as constants for all the media. These settings are suggested by the experts, and their description is beyond the scope of this report. The D_{50} illuminant [39] is used in the setup. The *observer angle* across all the measurements is 2^o at the temperature of 23^oC at measurement mode 2 (or M2) of the device.

Each category of medium has a different color profile. The default setting for each category is used in the experiment as suggested by the experts. These settings include the TAC limit, color grip, print speed, temperature, and super-heated steam.

5.5 Finding the Stabilization Time

Previous study show that the color keeps changing for a number of days. However, the rate of change of color decreases with time. We term the color to be dried if the change of color from that point of time is very little (or not observable by human eye). We find the minimum stabilization time so that the same model can be applied to the data that is collected after this time interval because the color has *not changed*. The aim of this study is to find the minimum time taken by the colors to stabilize. That is, the minimum time after which the change of color is *not significant*. To conduct this study, the measurements of $L*$, $a*$, $b*$ are taken at certain intervals of time, and the δE is computed with respect to the initial measurement. Hence, we get the δE values for that point of time.

The previous study in the organization showed that the color stabilizes after 24 hours. However, the study also suggested that this time interval varied significantly by media. Moreover, the data set was limited. Hence, it is important to determine if the color has stabilized before we determine the δE at that point of time. In Figure 5.2, the change of δE_{95} is < 0.1 after 24 hours. The trend is similar is case of other media used in the research as well.

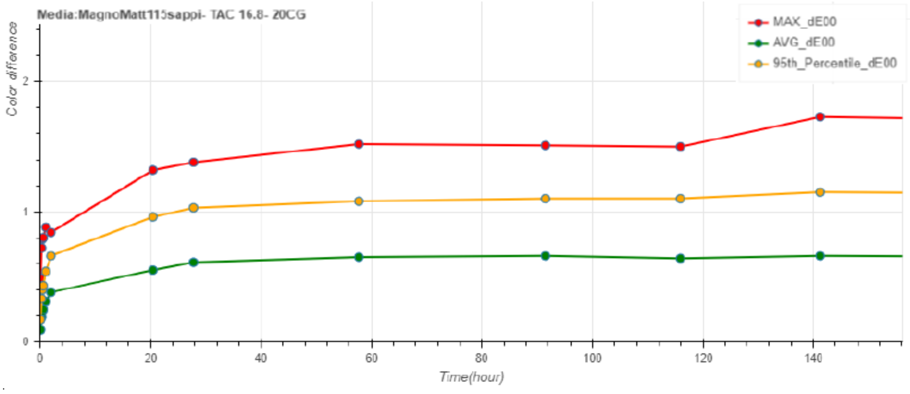


Figure 5.2: δE of Sappi Magno Matt Over Time

5.6 Preprocessing

The precise steps of preprocessing varies depending upon the algorithm in question. For example, the neural networks provide the best accuracy if the inputs, and outputs belong to certain range. However, this is not the case with tree-based algorithms. Moreover, the neural network does not accept categorical data (directly), whereas the categorical data can be directly used on tree-based algorithms. We discuss the distribution of data in Section 5.6.1, followed by analysis, and encoding of categorical data (in Section 5.6.2). Further, we study the outliers, and finally explore the different scaling techniques.

5.6.1 Distribution of Data

We discuss the distribution of data as this analysis would suggest the range, and distribution of each of the features which may or may not reveal biases in the data set. Such a possibility cannot be ruled out as the data set is relatively small, and covers limited numbers of suppliers, and media.

5.6.2 Categorical Data

The media in our data set has 5 different suppliers, with 3 coatings from 2 regions. The suppliers are Canon, Gprint, Sappi, UPM, and Verso. They have 3 coatings viz. Gloss, Silk, and Matt. The 2 regions are EU, and US. We observe that these quantities are categorical variables, and hence cannot be used directly on some regression techniques such as neural networks. Hence, they need to be encoded before they can be used in the model. The following two encoding algorithms were considered for this task:

- **One-hot Encoding:** In one hot encoding, each category is represented by a string of binary digits. This is useful when there is no relationship between the categories [9] [56]. In our dataset, the three independent categorical variables do not have any relationships between themselves. The representation of one hot encoding for the suppliers are listed in Table 5.2.

Supplier	**Encoding**
Canon	0 1 0 0 0
Gprint	0 0 1 0 0
Sappi	0 0 0 1 0
UPM	0 0 0 0 1
Verso	1 0 0 0 0

Table 5.2: One-hot Encoding of the Suppliers

- **Ordinal Encoding:** In ordinal encoding, each category is assigned an unique integer value. The SciKit Learn [47] ordinal encoding scheme arranges the list of categories in alphabetical order before assigning the integer value. The advantage of this algorithm is that it is the simplest representation of categorical data. However, there is a drawback in using this algorithm. It may provide a sense of relationship between categories (similar to the Likert Scale) when there is none [7]. For instance, refer to the Table 5.3. It provides the list of the category of suppliers along with their encoded values. Since, 1 is close to 0 than it is to 4, when these encoded values are used as input to the MLP regressor, the regressor would learn that the *Gprint* is closer to *Canon* than *Verso*. However, there is no evidence to suggest that the dry back pattern of *Gprint* is more similar to *Canon* than that of *Verso*.

Supplier	**Encoding**
Canon	0
Gprint	1
Sappi	2
UPM	3
Verso	4

Table 5.3: Ordinal Encoding of the Suppliers

5.6.3 Outliers

An outlier in a distribution is an observation that appears to deviate significantly from other observations. They could be a result of errors in measurement or a property of the data. Some outlier tests detect the presence of a single outlier (such as Doornbos' Test [15] and Grubbs' Test [22]) while other tests are able to detect multiple outliers (such as Hampel's Test [40] and Tukey's Test [59]) as it is not feasible to apply a single outlier test sequentially to find multiple outliers. The easiest way to detect outliers is to perform the Tukey's Test [59] (using the boxplot). A boxplot shows the distribution, and the skewness of the data. The box goes from the 25^{th} quantile to 75^{th} quantile. Hence, the whiskers at the end signify the top, and bottom 25 percentile. The points beyond these whiskers are the outliers. The distance of these points from the *box* is at least 1.5 times the IQR.

5.6.4 Scaling

Some of the regression models under consideration such as neural networks need the inputs, and outputs to be scaled. As the features in the real data can have a wide range of data values, models work better if we scaled all these data points [50].

Robust Scaler

The Robust Scaler removes the median, and scales the data to the Inter Quartile Range (IQR). The IQR is the range between the 25^{th} quantile, and 75^{th} quantile. The quartile range can also be customized based on the requirement. The median and IQR are computed for each feature and are stored so they can be used later to transform data [47]. This scaling technique is *robust* to outliers. This technique is an easy way is to remove the mean, and scale it to the unit variance.

However, if the data contains outliers, it can potentially influence the sample mean/variance in a negative way. In such cases, the median, and the interquartile range often give better results [47]. In our dataset, except for few features most of the features do not contain outliers. The SciKit Learn [47] library was used for the implementation.

Standard Scaler

The standard scalar also standardizes each feature independently by removing the mean, and scaling to unit variance. The standard score of the sample is calculated as $z = (x - u)/s$, where

u is the mean of the training samples, and s is the standard deviation of the training samples. In this technique, unlike robust scaler (which uses median and IQR), the mean and standard deviation are stored to be used on later data using the transform method [50]. This technique is most effective when machine learning estimators are sensitive to non-normal distributions and the individual features are not close to standard normal distribution [47]. The SciKit Learn [47] library was used for the implementation.

Normalization

In normalization, each row in the dataset (with at least one non-zero component) is rescaled independently (of other columns) so that its L_1 norm (or L_2, L_∞) equals one. Hence, it normalizes samples individually to unit norm. However, a drawback is that it cannot be inverse transformed (or *denormalized*) back to the original scale [47]. We need to have a scaling method that has the feature to inverse transform because the predicted values of $L*$, $a*$, and $b*$ are needed to be in original scale to compute δE. The SciKit Learn [47] library was used for the implementation.

Min-Max Scaler

This is a scaling technique that does not affect the distribution of the data. This algorithm scales down each feature individually within the specified range of the training set. In our case, the data is scaled between 0, and 1. SciKit Learn [47] Min-Max scaler library is used for the implementation. The advantage of min-max scaler is its reversibility. The predicted values (i.e., the $L*$, $a*$, and $b*$ values) can be scaled back to the original scale to calculate the δE. Due to these advantages, we use the minmax scaler for the Neural Network regression models.

5.7 Dimensionality Reduction

The dataset has 35720 ($47 * 760$) rows, and has a total of 55 dimensions. However, it is important to note that NOT all the dimensions are relevant to our study. Moreover, there are 6 dimensions for which adequate data is not available (discussed in detail in Section 5.3.2). Even if we keep that data aside, having such a large number of dimensions in the feature space would unnecessarily increase the volume of that space. It can expand to the extent that even a large amount of data would look sparse in the feature space. It is popularly called as the “Curse of Dimensionality”. It can have adverse effects on the performance of regression models. Trunk et al. in 1979 [57] discuss the concept of *Hughes Phenomenon*, where the power of regressor increases as we add the number of relevant features . However, after a certain number of dimensions, the performance starts to decline [8] [57]. Hence, there is a need to study the relevance of each of the feature, and use dimensionality reduction to bring down the dimensions. In other words, we need the feature elimination, and feature extraction at the same time.

5.7.1 Principal Component Analysis

Principal Component Analysis (PCA) reduces the dimensionality of a dataset with a large number of correlated variables, while retaining as much variation (or information) as possible present in the data set. Hence, this algorithm falls under the category of feature extraction. The variables are transformed into the principal components (PC), which are uncorrelated, and are in decreasing order of the variation present original variables. *Note:* input to the PCA is the raw data (and not the scaled data) as PCA itself performs the necessary scaling.

The motivation behind the use of PCA lies in the fact that it is a feature extraction method. Since there are a lot of candidate features that can be used in the regression model, we investigate the possibility of using fewer features without compromising with the accuracy during the implementation of the regression algorithms discussed later in this chapter.

The PCA calculates the principle components that capture the information (or variation) in the data. The principal components are determined in such a way that the first principal component signifies the largest possible variance in the dataset, the second principal component signifies the second largest variance in the dataset, and so on. Thus, the number of principal components can be chosen based on the information they capture. Hence, PCA is useful when we want to decrease the number of features used in the training data set. There are some disadvantages of PCA as well. The features provided as output by the PCA are not interpretable. This means that it will not be possible to study the significance of each of the original features in the regression model. There is always a loss of variation (or information) while choosing the number of PCs. However, if enough PCs are selected for training, this loss is not significant.

5.7.2 t-Distributed Stochastic Neighbor Embedding

Introduced by Maarten et al. [61], t-Distributed Stochastic Neighbor Embedding (tSNE) is another technique for reducing the dimensions, and visualizing the high-dimensional data sets. The prime difference between the tSNE, and PCA is that the tSNE preserves small pairwise distances (Equation 5.1)or local relations whereas PCA primarily preserves the large pairwise distances to maximize variance. The algorithm calculates a similarity score between pairs of features in the high dimensional space, and low dimensional space. Then, it optimizes these scores using a cost function. These high-dimensional data points (i.e., x_i, and x_j are converted into a joint probability distribution P over all pairs of non-identical points [62]. Hence, P is a matrix with the following entries in Equation 5.2 [62].

$$\delta_{ij}^2 = \|x_i - x_j\|^2 \tag{5.1}$$

$$p_{ij} = \frac{exp(-\delta_{ij}^2)/\sigma}{\sum\limits_k \sum\limits_{l \neq k} exp(-\delta_{ij}^2)/\sigma}, \forall i \forall j : i \neq j \tag{5.2}$$

t-SNE also defines q_{ij} that measures the similarity of the points y_i, and y_j in the low-dimensional space [62]. In other words, q_{ij} is the low-dimensional counterpart of p_{ij}. Hence, we get Equation 5.3 [62].

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum\limits_k \sum\limits_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}}, \forall i \forall j : i \neq j \tag{5.3}$$

First, we measure similarities between the points in the high dimensional space. For each point (in 2-d space) we enter a Gaussian distribution over that point, and measure the density of all points in that distribution. Now, normalize for all points to get the probabilities for all the points. So, higher is the similarity, higher is the probability. Now, instead of using a Gaussian distribution we use the Cauchy distribution. It is the Student t-distribution with one-dimension.

Hence, we have the second set of probabilities in the low dimensional space. Then, we measure the difference between the probability distributions of these two-dimensional spaces (i.e., the low dimension space, and the high dimension space) using Kullback-Liebler divergence [60](Equation 5.4). Finally, we minimize the Kullback-Liebler cost function (Equation 5.4) by using the gradient descent. The t-SNE algorithm was implemented using the Scikit Learn[47] library.

$$C(Y) = KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{5.4}$$

We will now discuss about the regression algorithms we use in the training. The tree-based regression algorithms and the SVR do not require any specific preprocessing. Furthermore, the prime features (i.e. $CMYK$ and $L*$, $a*$, $b*$), and *weight* do not have any missing value. However, there are few missing values for *opacity* and *gloss*. The average values of *opacity* and *gloss* from other media with the same coating have been used. There are few algorithms such as the DNN and MLP that require preprocessing (such as encoding and scaling) of data for optimal results. Hence, for these algorithms the categorical data is first encoded using ordinal encoding and then the complete data set is scaled using the Min-Max scaler. Now, our data is ready for regression analysis. In the subsequent sections we discuss the different regression techniques such as Neural Network (Section 5.8), ensemble methods (Section 5.10), and Support Vector Machines. In this situation, the $L*$, $a*$, and $b*$ at Time 96 can be independently predicted, and later used to compute the δE. In the subsequent sections, we briefly discuss the background of each of the regression algorithms used in this report. We have categorized it into 3 parts viz. neural networks, tree based, and support vectors machine. The tree based regressor are further divided into 2 categories: single decision trees, and ensemble learning algorithms. The hyperparameters are tuned to obtain the most optimal set of results from each of the algorithms. This is performed using the SciKit Learn [47] *GridSearchCV* that performs an exhaustive search over specified values for a given estimator.

5.8 Multi-Layer Perceptron (MLP) Regression

The result from Grid Search CV [47] suggests that a fully connected Multi-Layer Perceptron (MLP) Regression model with a single hidden layer of 20 neurons has least errors. Furthermore, it has 11, and 3 neurons at the input, and output layers respectively. Figure 5.3 shows the network diagram of the model. The SciKit Learn [47] library of *MLPRegressor* was used in the implementation.

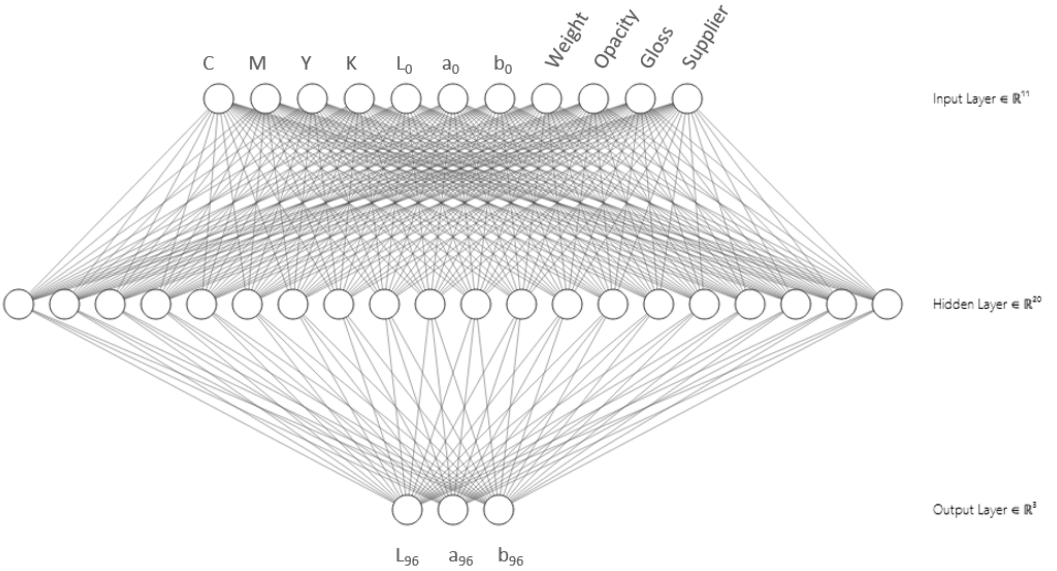

C
M
Y
K
L_0
a_0
b_0
Weight
Opacity
Gloss
Supplier
Input Layer ∈ $\mathbb{R}^{11}$
Hidden Layer ∈ $\mathbb{R}^{20}$
Output Layer ∈ $\mathbb{R}^{3}$
L_{96}
a_{96}
b_{96}

Figure 5.3: The MLP neural network model

5.8.1 Feature Importance

Analysis on the features show that the some features play a very significant role compared to others. Since, the MLP regressor model object does not have any attribute for feature importance, the *mlxtend* library [49] is used for analysis of feature importance. We first train the model on the training dataset, and estimate the baseline performance on the validation dataset. Now that we have the scores with all the features as input, we pick the feature in question, and perform the random permutation. Now, train the model with the *new* dataset. Then record the scores with the validation dataset. The difference between the scores of the model with original training data, and the scores of the model with modified dataset is the score for the feature importance. Now, perform the same exercise with all the features. A comparison can be performed between each feature with its scores.

5.9 Deep Neural Network (DNN)

In the previous section, we discussed, and evaluated the results of a simple MLP model with a single hidden layer. In this section, we pay more attention to the categorical variables. In the previous chapter, we had discussed the dangers of using the ordinal encoded values directly into the MLP regressor. Hence, there is another approach that can be used if the dataset contains both numerical, and categorical values. In this approach, the numerical values are directly fed to the dense layer. However, the categorical values are first encoded, and passed though the embedding layers. We define an embedding layer corresponding to each categorical variable. The model is visualized in Figure 5.4

In the DNN model, there are 4 input layers, 3 for each of the categories, and 1 for all the numerical variables. The input layer is connected to the dense layer with 18 neurons. The input layers for the categorical variables are connected to their respective Embedding layer. Each of these Embedding layers are connected to the Flatten layer whose outputs are concatenated to each other, and then again concatenated to the output of the other dense layer. These concatenated outputs are concatenated outputs are connected to the dense layer with 12 neurons. The output of this layer is finally connected to the output layer with 3 neurons (for 3 outputs). The loss function, and the metrics used are the MAE.

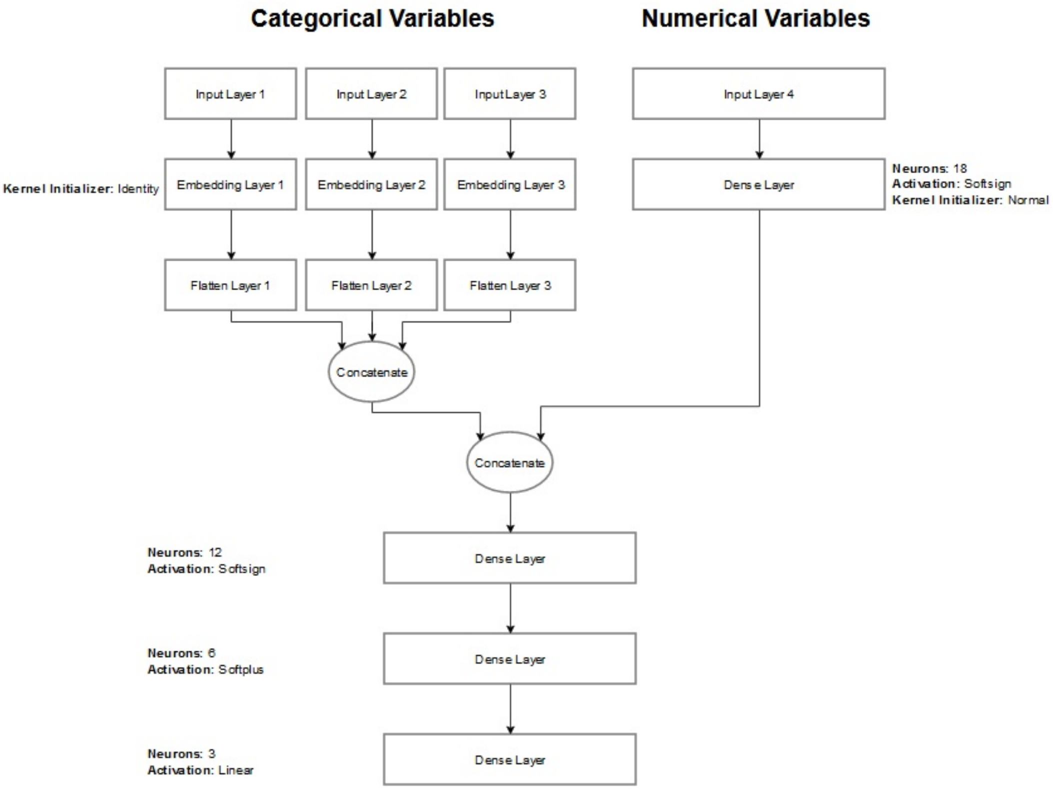


Figure 5.4: Deep Neural Network Model Architecture

5.10 Tree Based Regression Techniques

In the above subsection, we saw the neural network based regression technique. However, there are some other state of the art rule-based decision tree, and tree-based ensemble methods that have been proven to perform well [24] [18] [46]. One of the advantages of using these models is that it is not restricted by the scale of the input, and output. Since, there is a significant variation in the scale of various features in the dataset, it was important to scale them down to an optimal scale for lower MSE. However, the tree-based algorithms do not have such limitations. Hence, it has a potential to predict $L*$, $a*$, and $b*$ values with better accuracy than the MLP model.

5.10.1 Decision Tree (DT) Regression

The SciKit Learn library [47] offers a lot of hyperparameters to work with. The *criterion* measures the quality of the split. In this case both the *gini*, and *entropy* do not make a difference. The *splitter* is set to best as it takes longer to compute but chooses the best split. The *max_depth* is set to none as deeper tree has higher accuracy. The *min_impurity_split* is the minimum number of samples required to split a node. It can be used to control overfitting. The *min_sample_leaf*, which is the minimum sample required to be at a leaf is also set to default of 1. The *min_impurity_decrease* is set to default of 0.

5.10.2 Ensemble Learning

Bagging

The following results are obtained on implementation of random forest, and extra trees algorithms, discussed in Section 4.3.2.

- **Random Forest (RF) Regression:** The Scikit Learn [47] library provides an elegant implementation of the random forest regression algorithm. The number of trees (or *n_estimator*) can be explored to find the least error without having overfitting as very large number of trees can lead to overfit. The *max_features* is set to *auto*. The *min_samples_leaf*, and *min_samples_split* is set to default of 1, and 2 respectively, and there is no limit for *max_depth* so it is also set to default of *None*.

- **Extra Tree (ET) Regression:** The Scikit Learn [47] library was used for the implementation of the extra tree regression algorithm. The hyperparameters of Extra Trees algorithm is same as that of random forest (because the algorithm itself is very similar). Hence, the tuning is only performed on *n_estimator*.

Boosting

Since, these algorithms do not support multivariate output. Hence, three models were implemented for prediction of $L*$, $a*$, and $b*$ separately. The parameters remain the same for all of them. The reported MSE, and MAE values are computed using the same method as previous models. The δE values are computed from the $L*$, $a*$, $b*$ predicted from each of the three models.

- **AdaBoost (AB) Regression:** The Scikit Learn [47] library does not have a lot of hyperparameters to tune for the Adaptive Boosting algorithm. We perform the tuning on *n_estimtor*, and *learning_rate* whereas the *loss_function* remained *square*.

- **Gradient Boost (GB) Regression:** The Scikit Learn [47] library provides a plethora of hyperparameters to work with. These hyperparameters can be divided into 3 categories based on their functions viz. tree-Specific, boosting, and miscellaneous parameters. These tree-specific parameters are ones that we discussed in the previous regression algorithms such as *max_depth*, *min_samples_split*, *min_samples_leaf*, and *max_features*. The *max_depth* is the maximum depth of each of the tree which is set to default (of 3) as increasing it did not improve the model but increased the training time. The *min_samples_split*, and *min_samples_leaf* are the same as discussed in the Random Forest. Hence, *min_samples_split* is minimum number of samples required to split a node in each tree, and *min_samples_split* is the minimum number of samples required at a leaf node [47]. These values are set to default of 2, and 1 respectively. *max_features* is the number of features that are considered while making a split at each node of a tree. Since, we do not have a lot of features, this parameter is set to consider all the features.

  There are some hyper-parameters that are specific to boosting algorithm such as *learning_rate*, *n_estimators*, and *subsample*. These parameters primarily set this algorithm (and boosting, in general) apart from other algorithms we explored. Hence, we explore different combinations of these parameters. The *learning_rate* determines the impact of each tree, hence can be used to enhance the performance. We also explore the different values of *n_estimators* to reach an optimal value. *subsample* is the fraction of samples that is used to fit the individual base (or weak) learners. If it is < 1.0, it is Stochastic Gradient Boosting [47]. However, results have shown no improvements on decreasing *subsample*. In the other miscellaneous hyperparameters, the *warm_start* is set to *True*, the loss function is set to *lad* (or least absolute deviation), and *criterion* is set to MAE.

- **Extreme Gradient Boost (XGB) Regression:**

  XGBoost open-source library [10] provides a robust implementation of the algorithm. The hyperparameters of this algorithm are very similar to Gradient Boosting algorithm. Hence, most of those parameters remain the same. However, there are some hyperparameters specific to XGBoost that need some discussion. We have *min_child_weight* that is similar to the *min_samples_leaf* in GB, but in case XGB, it is the minimum sum of weights. We also have alpha, and lambda which are the L1, and L2 regularization terms respectively. The objective (loss) function is set to *reg:linear*.

> The parameter *grow_policy* controls the way child nodes are added to the tree. The *depthwise* split makes the split at nodes closest to the root, while the *lossguide* split makes the split at nodes with the maximum change of loss [10].

Most of the well-established regression models, are complex, and require high computational resources during training [17]. Rule-based decision tree, and tree-based ensemble methods, e.g., Gradient Boosting, Random Forest, and eXtremely Randomized Trees (Extra-Trees) have recently begun to gain momentum, because they are simple but still powerful, and robust predictive algorithms, and they have less parameters for tuning [24] [18] [46].

5.11 Support Vector (SV) Regression

There are primarily 4 hyperparameters in Scikit Learn [47] that determine the accuracy of a support vector model. The regularization parameter (or C), *kernel*, *degree* of the polynomial kernel function,, and kernel coefficient (or *gamma*). The degree of polynomial is only relevant if the kernel is *poly*. The *gamma* is set to default of $frac1n\_features$. The hyperparameter tuning is performed to determine the *kernel*, and *C*.

Just like the boosting regression algorithms, Support Vector Regression does not support multivariate output. Hence, three different models (with same parameters) were implemented for prediction of $L*$, $a*$, and $b*$ values. The reported MSE, and MAE values are computed using the same method as previous models. The δE values are computed from the $L*$, $a*$, $b*$ predicted from each of the three models.

5.12 Validation

In the previous sections, we discussed the approaches for implementing the model, and testing it. However, the model(s) will be put to test when a new medium is used for prediction, and compared with the actual values of $L*$, $a*$, and $b*$ at Time 96. Since, the main aim of this project is to assist the operators to make high quality color profiles without costing a significant amount of time, this model will be put to use only with new media, since color profiles already exist for existing media. Hence, it is important to evaluate the performance of this model with new medium (i.e., a medium that was not used for training).

We conduct 2 set of validation tests, one from our perspective, and another from the perspective of the use case. Once we have compared the models, and selected the model to use, we retrain the model with almost the same data set but remove a medium this time. The removed medium is used for evaluating the model as it is “unknown” to the model. The test is done iteratively by removing different medium from the data set each time, and retraining the model.

The other validation test has the setup of the actual use case, i.e., to make a color profile. Currently, the operators of the printers use the default media entry available based on the coating for color profiles. This does not produce the best quality results but saves the time. Hence, the dried measurements of $L*$, $a*$, $b*$ is used as a reference for comparison. The color profiles are made from $L*$, $a*$, $b*$ values using the *prismaSync* software developed by the color management team at CPP. If the color profile made by the predicted values is close to the actual color profile (made using $L*$, $a*$, $b*$ values at Time 96), we conclude that the model performs sufficiently well, and can be used to make color profiles for new media. In addition to the color profile, the δE values can also be computed and compared with the test results. We use new media for this test, and the details are as follows.

Serial Number	**Supplier**	**Medium Name**	**Weight**	**Coating**	**Region**
1	Gold East	Space Shuttle	157 gsm	Matt	CN
2	Zanders	Silver Digital	200 gsm	Matt	EU
3	Zhonghua	Ninbo Star	250 gsm	Gloss	CN
4	Sappi	Magno Plus Gloss	150gsm	Gloss	EU

Table 5.4: Media for Validation

5.13 Summary

In this chapter, we discussed the entire implementation phase of the project starting from the selecting the media, collecting the data, preprocessing it, and implementing the regression models. Each of these algorithms have their own set of parameters that need to be tuned to get the best results. In the next chapter, we discuss the results obtained on running the aforementioned algorithms under different parameters.

Chapter 6

Results

In Chapter 5, we discussed about implementations involved in various stages of this project. In this chapter, each section is dedicated to the results obtained upon implementation of those algorithms/tests. In Section 6.1, we cover the results obtained from the measurement accuracy test. Further, in Section 6.2, we discuss the discuss the results obtained from the tests conducted during the data collection phase. Section 6.3 details some interesting results obtained upon some tests and preprocessing on the collected data. Section 6.4 describes the data obtained on applying the dimensionality reduction algorithms. In the subsequent sections of this chapter, we discuss the results obtained on implementing the MLP regression model (Section 6.5), deep neural network regression model (Section 6.6), tree based regression algorithms (Section 5.10) and support vector regression (Section 6.8)

6.1 Measurement Accuracy Test

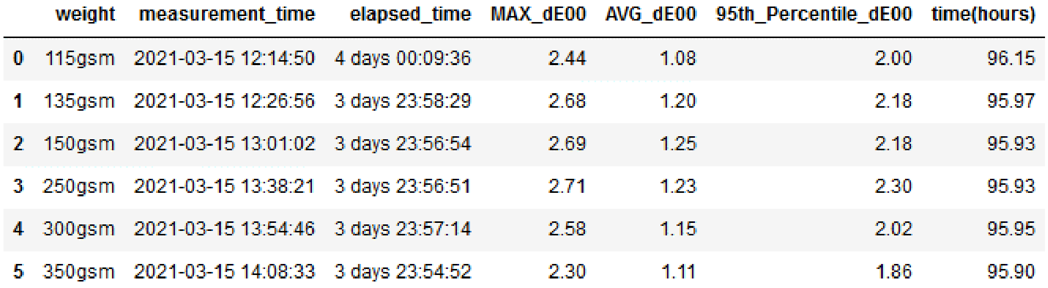
The setup of this test was discussed in Section 5.1. The results suggest that the standard deviation, average, maximum and 95 percentile are about 0.06, 0.09, 0.42 and 0.2 respectively. The prediction model cannot predict something with higher accuracy than its measurement as it is hard to predict something that cannot be measured. These numbers set a limit to the prediction model. The results have been deemed satisfactory by the color experts in CPP and other spectrophotometers used in the organization have also shown similar margin of error in the past. The table of results of the accuracy test is provided in Appendix B.

6.2 Data Collection

In this section, we discuss the results obtained on performing some tests while collecting the data. These tests help us to verify if the target accuracy for the model is realistic and can be achieved (with sufficient data and robust model). These tests also help us to find the time interval after which it can be said with certain level of certainty that the ink in the media has dried (or stabilized).

6.2.1 Effect of Weight on Dry Back

The motivation and setup of the experiment were discussed in the Section 5.3. Figure 6.1 suggests that the δE_{avg}, δE_{max} and δE_{95} for each weight are relatively close to each other. The range of these values is 0.17, 0.4 and 0.4 respectively.



	weight	measurement_time	elapsed_time	MAX_dE00	AVG_dE00	95th_Percentile_dE00	time(hours)
0	115gsm	2021-03-15 12:14:50	4 days 00:09:36	2.44	1.08	2.00	96.15
1	135gsm	2021-03-15 12:26:56	3 days 23:58:29	2.68	1.20	2.18	95.97
2	150gsm	2021-03-15 13:01:02	3 days 23:56:54	2.69	1.25	2.18	95.93
3	250gsm	2021-03-15 13:38:21	3 days 23:56:51	2.71	1.23	2.30	95.93
4	300gsm	2021-03-15 13:54:46	3 days 23:57:14	2.58	1.15	2.02	95.95
5	350gsm	2021-03-15 14:08:33	3 days 23:54:52	2.30	1.11	1.86	95.90

Figure 6.1: Overview of δE of the Weight Categories

However, this test does not conclusively suggest an outcome. The δE for each weight group has a distribution with 760 data points. Hence, if the distributions are the same, then it can be concluded that the weight does not have any impact. In this data set, we have 6 distributions of δE, each distribution corresponds to a weight category.

To determine if the two distributions are the same we used the Kolmogorov-Smirnov Test [38]. Kolmogorov-Smirnov Test [38] is used to find if the given 2 sets of data belong to the same distribution i.e. the two distributions of the those points are equal. The 2 sample Kolmogorov-Smirnov Test [38] has the following assumptions:

- The two samples are independent.
- The outcomes are ordinal or numerical.

The above assumptions are satisfied by the given dataset. Since we have 6 distributions (by weight), a pair of distributions is selected and the test is conducted under the following hypothesis for 2 sample Kolmogorov-Smirnov Test [38].

H_0 : The two dataset values are from the same continuous distribution.
H_1 : The two dataset values are NOT from the same continuous distribution.

The test suggests that the almost none of the distributions are the same except for the 135gsm and 250gsm. However, when a similar test was performed with the data from the Measurement Device Accuracy Test (Section 5.1), the result was similar. The distributions of the δE from the same device from the same medium under same conditions were not the same. The setup of that study is provided in the Appendix C. A plausible reason is that the measurement error is significant enough to change the distribution of each set of measurement under the same setting.

Since, weight does not have a significant impact on the drying effect, the data of a wide range of different weights of media was not required to be collected for each medium. Since, not all weights have an impact, some of the weight categories are not included in the dataset as well.

6.2.2 Finding the Stabilization Time

The motivation and the experimental setup to find the stabilization time was discussed in Section 6.2.2. In this subsection, we discuss the results and its implications. The study in the organization had shown that the drying time is about 24 hours. However, when the same study was conducted with the new data, there were some media that took longer than 24 hours to stabilize. In Figure 6.2, the δE 24 hours and 96 hours are listed for each medium used in the prediction model. The Digi Finesse Premium Silk by UPM showed significant difference in δE between 24 hours and 96 hours.

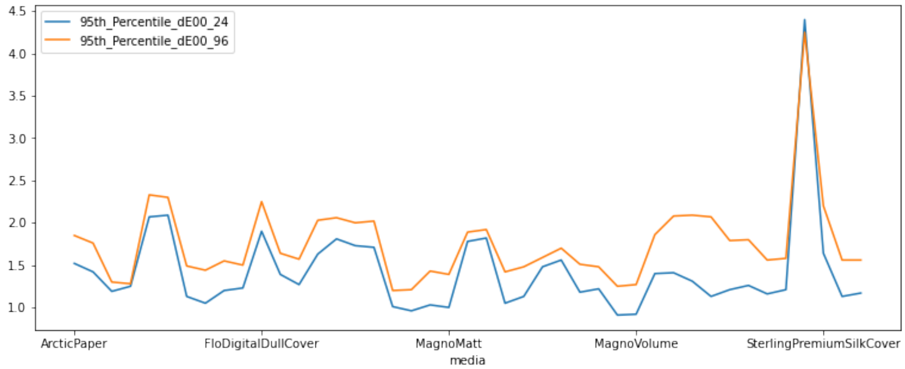


Figure 6.2: δE After 24 Hours and 96 Hours

The colors (for each medium) have fairly stabilized after 96 hours. Hence, the δE to be used in the prediction model is between the initial $L*$, $a*$, $b*$ (hitherto called Time 00) and the $L*$, $a*$, $b*$ measurements after 96 hours (hitherto called Time 96).

6.3 Preprocessing

In this section, we discuss the results of the analyses performed in the pre-processing phase, discussed in Section 5.6.

6.3.1 Distribution of Data

The histograms in Figure 6.3, Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8 represents the frequency of each of those discrete values.

These histograms suggest that these variables are discrete and only take few values. The $CMYK$ are obtained from the test chart and the Opacity and Gloss values are the media characteristics obtained from the respective suppliers.

The plots in Figure 6.10, Figure 6.11 and Figure 6.12 represent the probability density functions and histograms of the measured dependent variables. The plots in Figure 6.14, Figure 6.15, Figure 6.16 represent the probability density functions and histograms of the independent variables

The distribution of each of the dependent and independent variables suggest some interesting outcomes. The distribution of $L*$, $a*$ and $b*$ before and after drying are very similar. This result

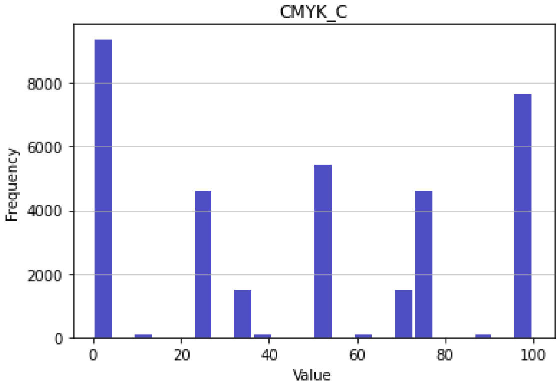


Figure 6.3: C

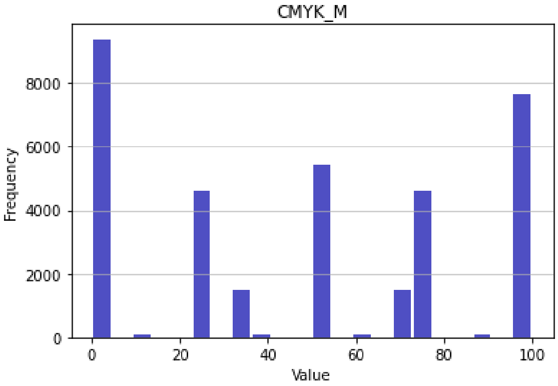


Figure 6.4: M

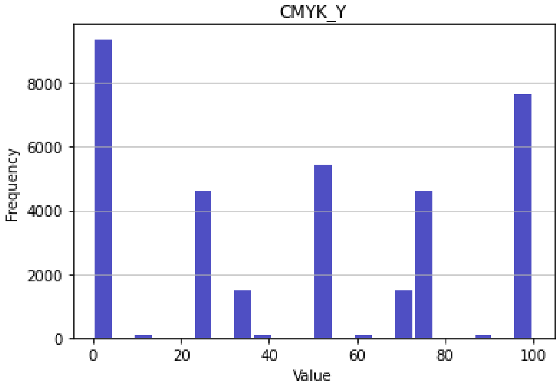


Figure 6.5: Y

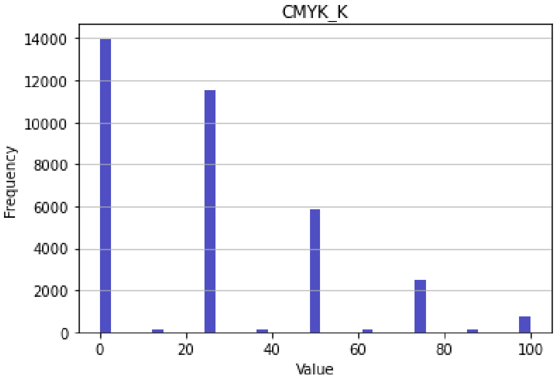


Figure 6.6: K

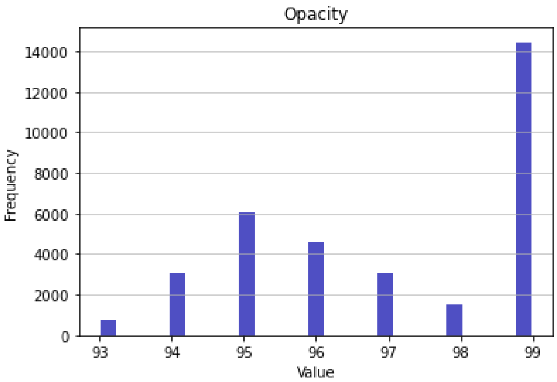


Figure 6.7: Opacity

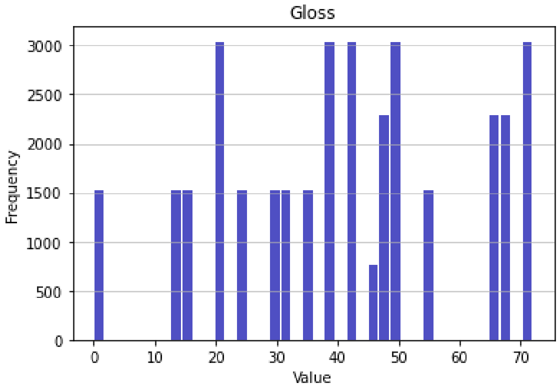


Figure 6.8: Gloss

Figure 6.9: Probability Density Function of the Independent Variables

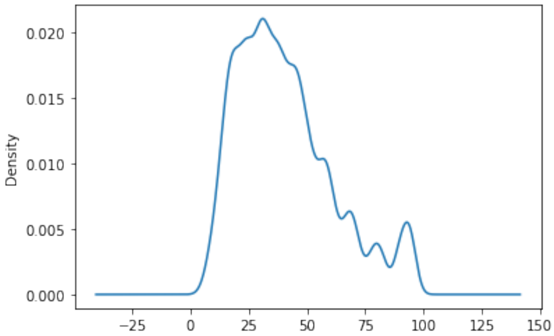


Figure 6.10: L

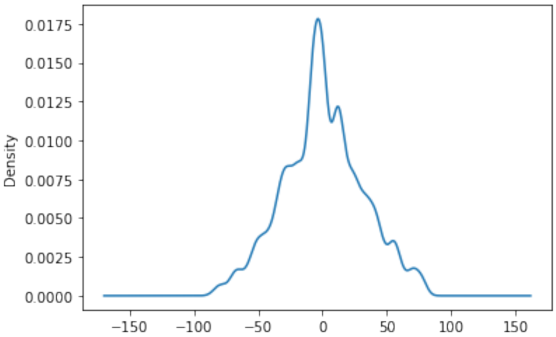


Figure 6.11: a

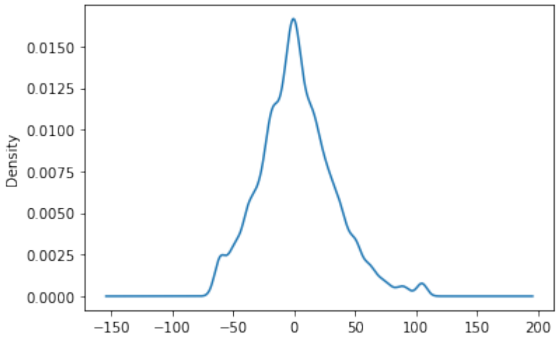


Figure 6.12: b

Figure 6.13: Probability Density Function of the Measured Independent Variables

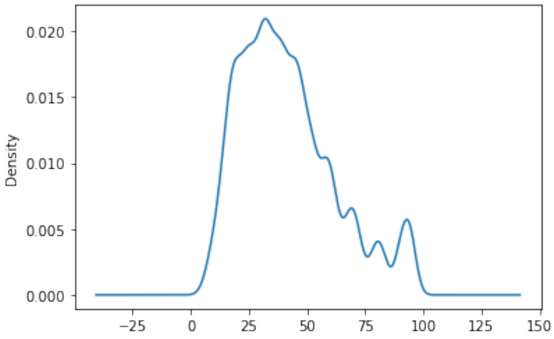


Figure 6.14: L_dry

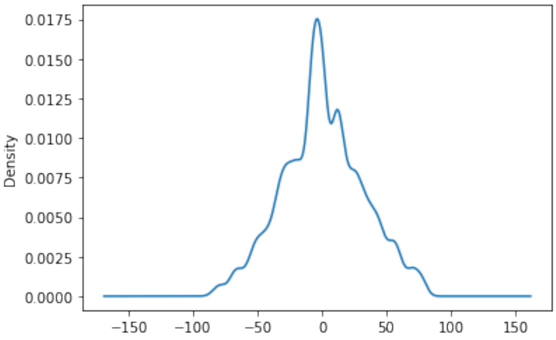


Figure 6.15: a_dry

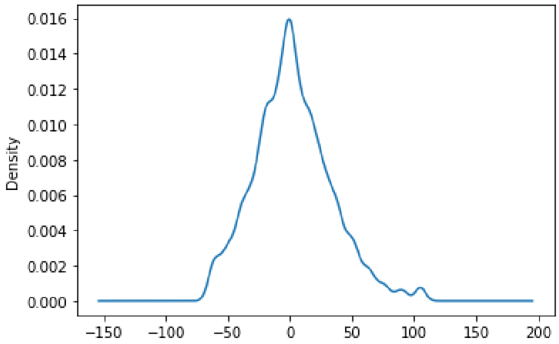


Figure 6.16: b_dry

Figure 6.17: Probability Density Function of the Target Variables

is coherent to the fact the correlation of the respective values of $L*$, $a*$ and $b*$ before and after drying are very high (See Appendix D).

6.3.2 Outliers

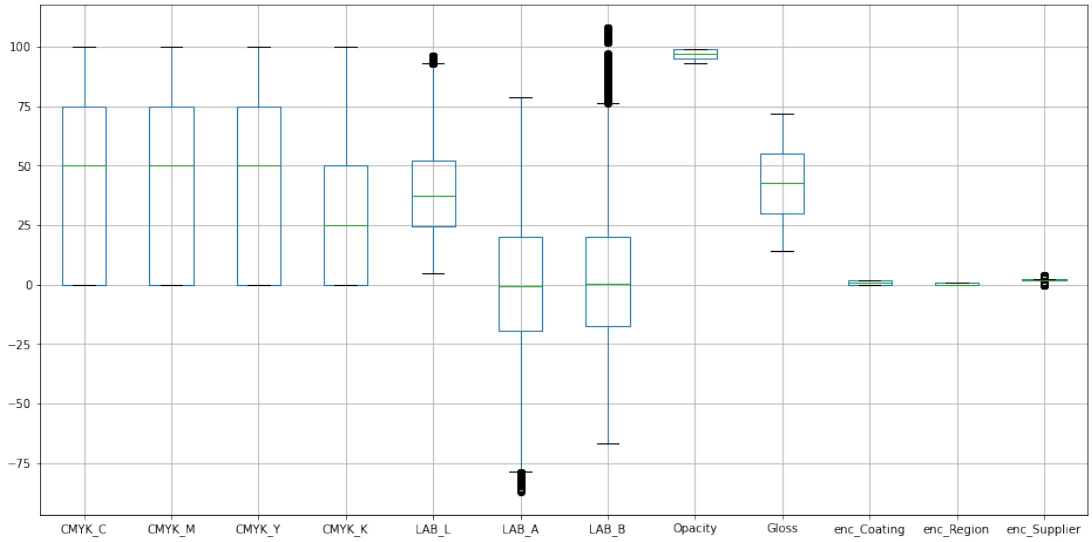


Figure 6.18: Boxplot of the Independent Variables

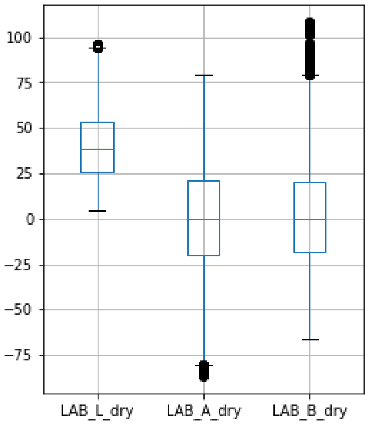


Figure 6.19: Boxplot of the Dependent Variables

The outlier's analysis performed in Section 5.6.3 suggests that the C, M, Y, K values are skewed towards the higher values (or right, if the plot is considered horizontal) with their median at 50, except K whose median is at 25. Perhaps, that is because the values cannot be negative, so it sets the lower bound for the values. The $L*$, $a*$, $b*$ values are spread out in their respective range of values i.e. $[0, 100]$, $[-127, 128]$ and $[-127, 128]$. There are a few outliers in $L*$, $a*$ and (encoded) suppliers. Moreover, there are a significant number of outliers in $b*$, compared to other quantities for unknown reasons. A similar trend is observed in the dependent variables (i.e. $L*$, $a*$, $b*$ values at Time 96) because there is a very high correlation with the $L*$, $a*$, $b*$ values at Time 00. This means the use of regression techniques that are sensitive to outliers should be avoided.

6.3.3 Scaling

We discussed the techniques for scaling in Section 5.6.4, i.e. Min-Max scaling, Robust scaling, Standard scaling and Normalization. In this section, we see the results when the data was preprocessed using the aforementioned techniques using the Scikit [47] library. Then it was used in MLP regressor to obtain the results in Table 6.1. **Note:** The δE in the table is computed between the actual $L*$, $a*$ and $b*$ values and the predicted $L*$, $a*$ and $b*$ values at Time 96.

	Min-Max	**Robust**	**Standard**	**Normalization**
Time Elapsed (s)	0.24	0.32	0.55	0.27
MSE	0.003	0.008	0.013	0.001
R2 Score	0.93	0.98	0.99	0.94
Avg δE	0.07	0.15	0.17	0.07
95% δE	0.16	0.31	0.34	0.13
Max δE	0.39	0.60	0.64	0.26

Table 6.1: Effect of different scaling algorithms on the output

The above techniques were applied to a smaller set (in fact, a subset with 3 media) to obtain the results in Table 6.1. In Table 6.1, it is observed that the time elapsed (to run the regression algorithm) is least for the min-max scaler and followed by normalizer, robust scaler and standard scaler. The key factors to determine the performance of the model are the MSE and the δE_{95}. In case of the MSE and δE_{95}, it is observed that the best results are obtained for the normalizer followed by the min-max scaler, robust scaler and standard scaler. Apart from the R2 score, all the other metrices follow the same trend. The standard scaling of the data does not significantly influence the results. However, as we observed in the previous sections, there are a few features that contain outliers. Since, MLP regressors are sensitive to outliers, this can be the reason for relatively better performance of the Robust scaling. Since, we obtained the best results with Min-Max, we use this scaling techniques for the neural networks.

6.4 Dimensionality Reduction

In this section, we will try to obtain the dataset that can be used later in the regression. We will compare the different algorithms and select one ones that yield the best dataset i.e. the ones that capture more information, save space, and computation time.

6.4.1 Principal Components Analysis

The concept of the PCA was discussed in Section 5.7.1. We have the 13 features (viz. C, M, Y, K, $L*$, $a*$, $b*$, Weight, Opacity, Gloss, Supplier, Coating, Region) under consideration for inputs to the model. When this algorithm was used to obtain the principal components (or PC), the results were as shown in Table 6.2.

Principal Component	**Explained Variation (%)**	**Cumulative Explained Variation (%)**
PC0	24.38	24.38
PC1	17.58	41.96
PC2	14.21	56.17
PC3	12.96	69.13
PC4	12.59	81.72
PC5	6.88	88.60
PC6	4.39	92.99
PC7	3.31	96.3

Table 6.2: Principal Components obtained from the inputs

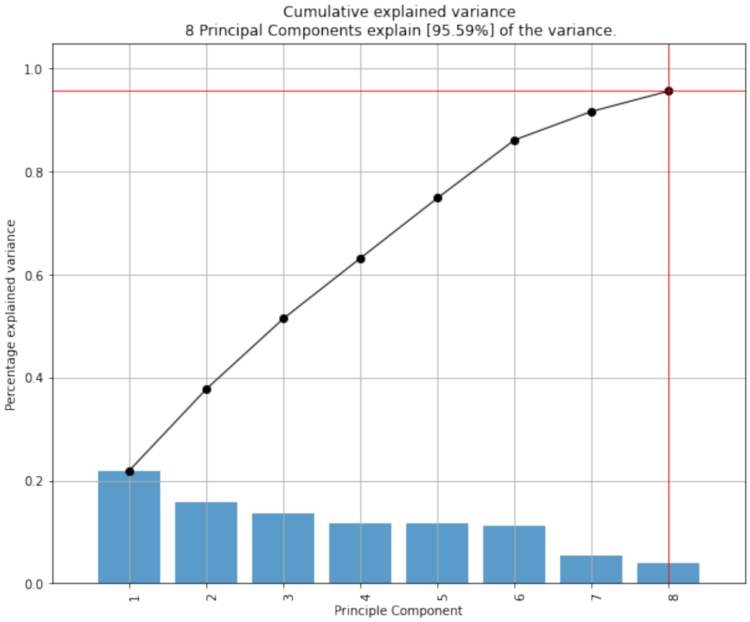


Figure 6.20: Percentage variance explained by each principle component

In Table 6.2 and Figure 6.20, we observe that more than half of the variation is explained by the three principal components. The eight principal components explain 96% of the variation in the data. The number of required features is down to 8 (from 13). This would mean a significant improvement in training time with information loss of about 3-4%. To obtain the results in the Table 6.3, we used 80% of the dataset for training and the remaining 20% for testing. We have only used two algorithms to study the impact of PCA.

	MLP (PCA)	**MLP (Regular Data)**	**RF (PCA)**	**RF (Regular Data)**
MSE	0.212	0.164	0.447	0.304
MAE	0.387	0.299	0.501	0.366
R2 Score	0.986	0.987	0.993	0.999
Avg δE	0.26	0.15	0.52	0.473
95% δE	0.84	0.751	1.738	1.166
Max δE	2.92	2.42	7.561	6.144

Table 6.3: Performance Comparison Between Each Regression Algorithm

In Table 6.3, we observe that when the features from the PCA were used in the MLP regression model, there was no significant difference in the errors and δE values. It should be noted that the MLP models were different for both the datasets. The MLP model for the regular features has 20 neurons in the hidden layer, whereas the MLP model for the PCA features has 13 neurons. We also observe a similar trend in case of random forest i.e. the similarity in performance of the model in both the cases. The random forest model was the same in both the cases.

6.5 Multi-Layer Perceptron (MLP) Regression

In Section 6.4, we discussed the algorithms to reduce the number of dimensions in use. We observed when we use PCA, we are able to reduce the number of dimensions from 13 to 8 while losing only 3-4% of data. However, we also observed that applying the PCA did not bring a substantial decrease in the training time. Hence, we choose not to use the PCA and include all the features in the training data. Note that there are three categorical features in the dataset. MLP regression technique cannot be directly used to evaluate the categorical features. Hence, they have been encoded using the ordinal encoding (Section 5.6.2).

The data set was trained on the MLP regressor with different activation functions and solvers. Figure 6.21 represents the heatmap with MAE for each of combination activation and solver used. The values of the other metrices is available in Appendix E.1.

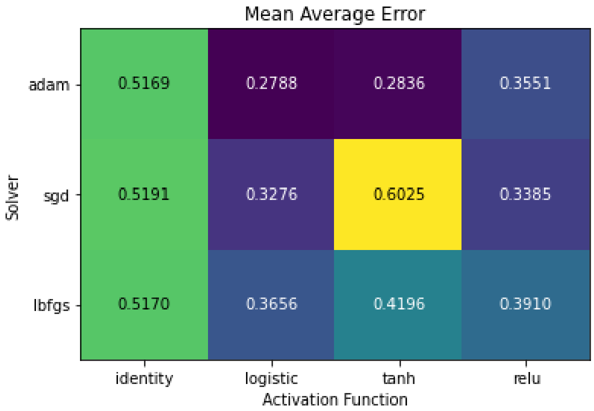


Figure 6.21: Heatmap of MAE for activation functions and solvers in MLP regression

The above heatmap suggests that the Adaptive Moment estimator (or Adam solver) with the Logistic activation function has the least MAE. A similar trend is also observed with other metrices such as MAE δE_{95}, δE_{avg}, δE_{max} as well.

We try to find the importance of each of the features to learn more about the factors that affect the drying behavior. Moreover, this will help us to potentially eliminate the less relevant features from the training data. However, the MLP regressor by SciKit Learn [47] does not have an inbuilt function to get the scores. Hence a different approach is taken which is discussed in the next subsection.

6.5.1 Feature importance

While predicting the $L*$, $a*$ and $b*$ values we also try to identify the features that contributed the most in the prediction. There could also be some features that are not very relevant to the prediction. The features used in the model were chosen based on the domain knowledge and their availability. Hence, we can expect to find some interesting results from the test on feature importance. However, the MLP regressor library by SciKit Learn [47] does not provide any function (or return any object) to evaluate the feature importance. Hence, we used the *mlxtend* library [48] to evaluate the importance of each feature.

The library takes a model that was fit to the training data and estimate the performance (by using the r2 score) of the model on the validation dataset. So, it now has the baseline performance. Then, it iterates over all the features and for each feature, it randomly permutes the column. Then, it records the performance (r2 score) of the model and compares it with the baseline performance [48]. This algorithm has a disadvantage that the importance of correlated features may be overestimated [55]. This would reflect on our results as we observed that there is a significant correlation between the respective values of $L*$, $a*$ and $b*$ at Time 00 and Time 96.

Figure 6.22 provides the importance score for each feature. It is evident that the Coating and Region are the lowest and insignificant. Hence, they are not being used in the dataset of the final model. However, the data set has 3 types coating and from just 2 regions. Since, there is not much variation in the encoded coating and region, we can expect the growth of their importance if more regions are included in the data set.

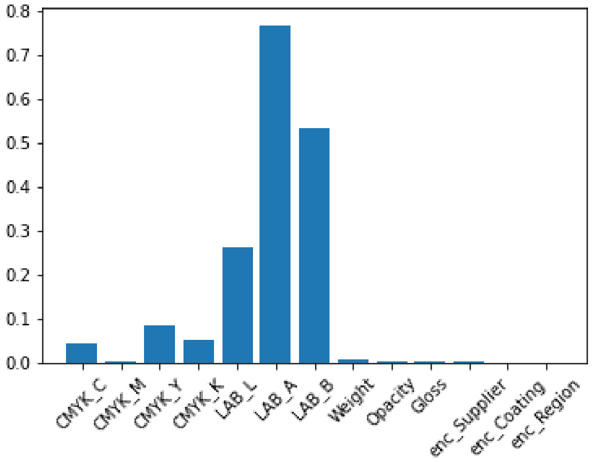


Figure 6.22: Feature Importance Plot with MLP regressor

6.5.2 Media Characteristics

In Section 5.3.2, we discussed several media characteristics and their limitations for using them in the model. However, it would be interesting to see these characteristics have enough information that can help in improving the accuracy of the model. If the media characteristics turn out to be useful, it can be a starting point for an approach to further improve the accuracy of the model. Since, there are a lot of missing values of the media characteristics. We changed the training data set to use only the media that had the media characteristics. Figure 6.23 is the plot of the feature importance score of the media characteristics obtained by using the *mlxtend* library [48]. The other features are not included in the model as these values are in different scale.

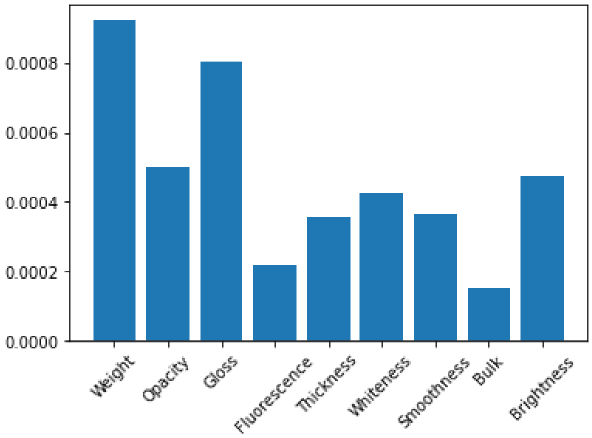


Figure 6.23: Feature Importance Plot for media characteristics

The above plot suggests that the Weight, Opacity, and Gloss have relatively high importance when compared to other media characteristics. Moreover, the brightness also has a high score. However, the since a lot of data is not available for brightness, it cannot be used for reliable results. However, if there is a need to increase the accuracy in the model by including more features in the future, brightness and whiteness are the best candidates for data collection.

We have selected the dataset to be used for each of the regression algorithm. Now, we discuss the results in the successive sections.

6.6 Deep Neural Network

While implementing the neural network we used different combinations of activation functions for the dense layers. The *Keras* library [12] provides a list of predefined activation functions such as *RELU, linear, sigmoid, softmax, softplus, softsign, tanh, exponential.* Various combinations of these activation functions were used in the models with *RELU* and *linear* functions in the output layer. Figure 6.24 and Figure 6.25 are the results when these activation functions were used in dense layers with linear function in the output layer.

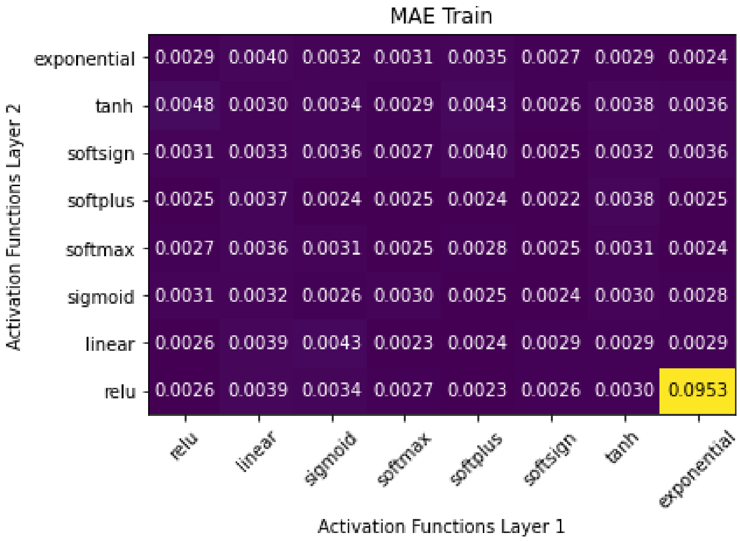


Figure 6.24: Heatmap of MAE on the Training Data with Linear Activation function in the last layer

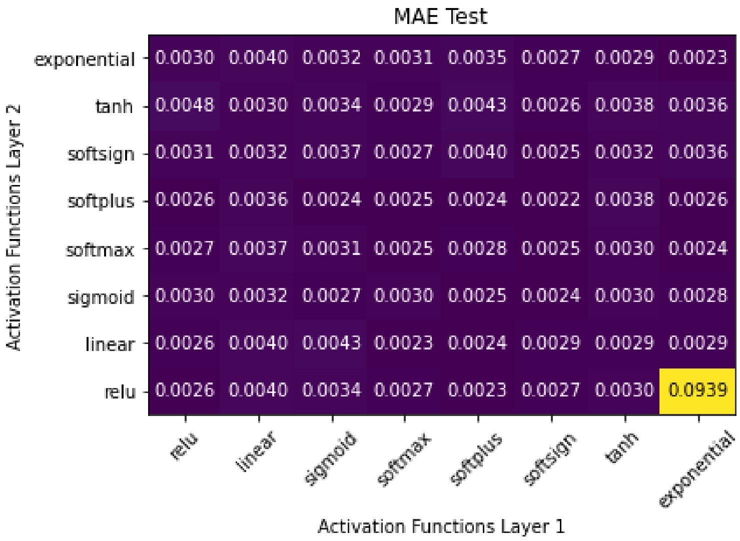


Figure 6.25: Heatmap of MAE on the Testing Data with Linear Activation function in the last layer

Similarly, Figure 6.26 and Figure 6.27 are obtained when *RELU* was used in the output layer.

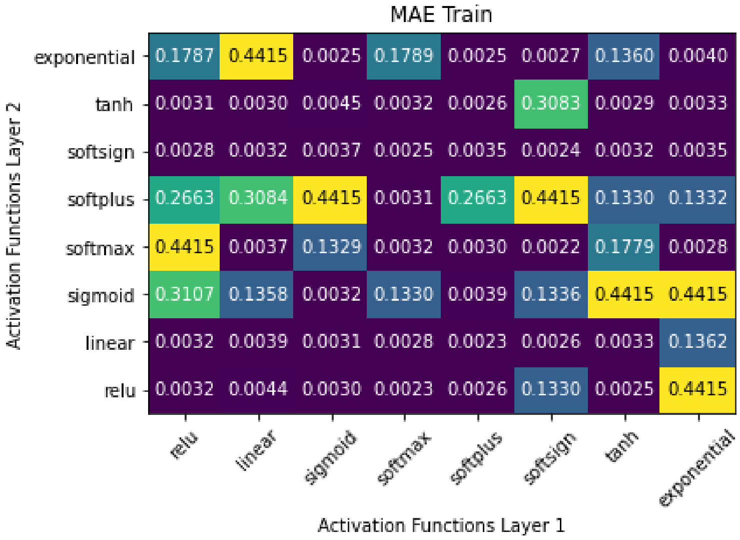


Figure 6.26: Heatmap of MAE on the Training Data with RELU Activation function in the last layer

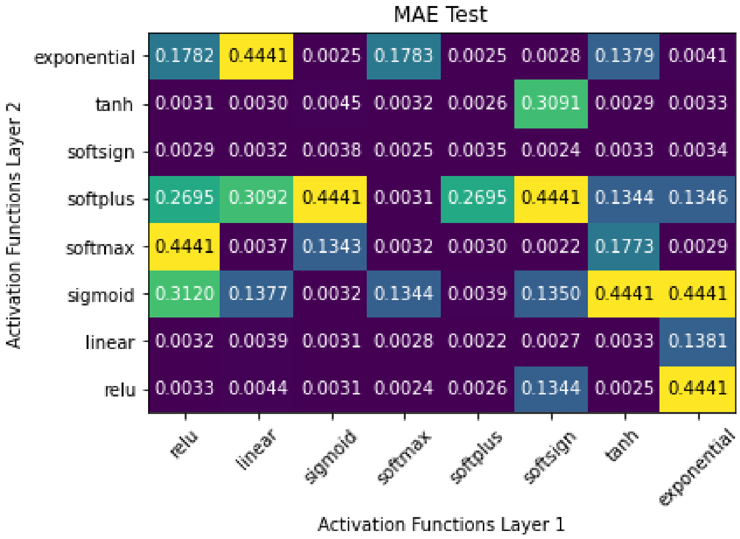


Figure 6.27: Heatmap of MAE on the Testing Data with RELU Activation function in the last layer

In Figure 6.24, Figure 6.25, Figure 6.26, Figure 6.27, we observe that the MAE values with the *linear* activation function is considerably lower than that of *RELU*. This can be attributed to the fact that the RELU never returns negative value, whereas $a*$ and $b*$ take negative values. Furthermore, we also observe that the *softsign* and *softplus* activation functions have the minimum MAE values for train and test data. However, other combinations such as *softmax-linear* and *softplus-softplus* some of the least MAE values. Hence, we also consider other metrics i.e. δE for analysis (Appendix E.2.1). There is a very small difference between the δE_{95} values for these combinations, but it is clear that the *softsign- softplus* combination is still the best.

6.6.1 Optimizers

The optimizers in the *Keras* [12] are the methods that change the attributes of the deep learning model such as weights and learning rate to reduce the losses (or the MAE, in our case). Hence, the right optimizers can reduce the number of epochs and help in quicker convergence. *Keras* [12] provides a wide range of optimizers (mentioned in Figure 6.28). The activation functions thus obtained in the previous subsection are used in these models and the results are as follows (Figure 6.28).

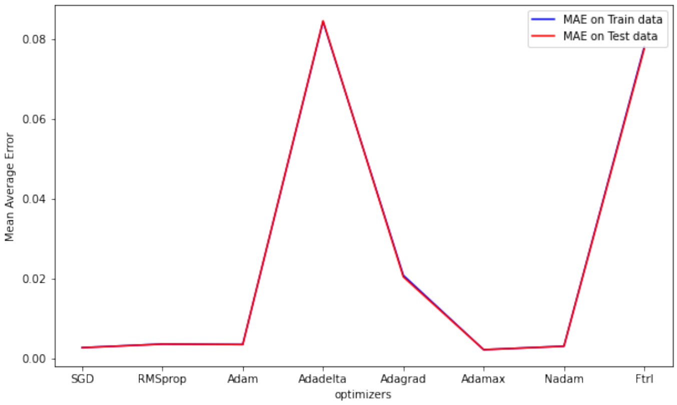


Figure 6.28: Plot of MAE on Train and Test Data with Different Optimizers

There are a number of optimizers that result in very low MSE. The lowest is in the case of *Adamax* and followed by *SGD* and *RMSprop*. There is no overfitting in the data. In fact, similar trend is followed in case of other metrices such as δE_{95} (Appendix E.2.2), MSE (Appendix E.2.2) etc. as well.

6.6.2 Embedding Initializers

The deep neural network has 3 embedding layers, each corresponding to one categorical variable. The optimally initialized embedding matrix would reach convergence relatively faster. *Keras* [12] provides 11 such initializers and the MAE values of their respective models are plotted in Figure 6.29.

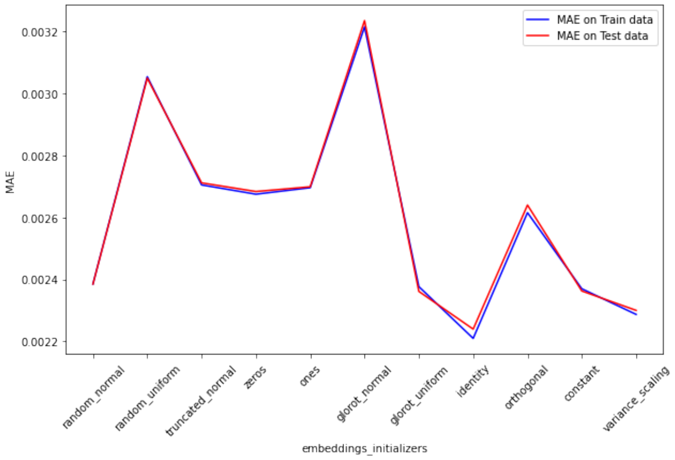


Figure 6.29: Plot of Errors and δE on the Training and Testing Data with Over Different Initializers for Embedding layers

In Figure 6.29, the *identity* has the minimum MAE. The metrices such as δE_{95} (Appendix E.2.3) and MSE (Appendix E.2.3) also have the same minimum. Moreover, we also observe that there is no overfitting.

6.7 Tree Based Regression Algorithms

The decision trees are inherently more intuitive and interpretable than the MLP models. Moreover, the decision trees are nt bounded by the scale and distribution of data. However, using a single decision tree might not be sufficient hence, we also use ensemble learning. The working principle of each of these techniques has been discussed in Chapter 5. The results for each of those algorithms are discussed in the subsequent subsections.

6.7.1 Decision Tree (DT) Regression

Decision trees are one of the most interpretable regression techniques, which has evolved into more complex techniques. An advantage of the decision tree is that it is simple and does not have a lot of parameters to tune. During the implementation (using SciKit-learn [47] library), we defined the *Mean Absolute Error (or MAE)* as the loss function. MAE minimizes the L1 loss using the median of each terminal node. The accuracy score and the r2 score were consistently very high (99.9%) for all the observations. Hence, they have not been included in the following tables.

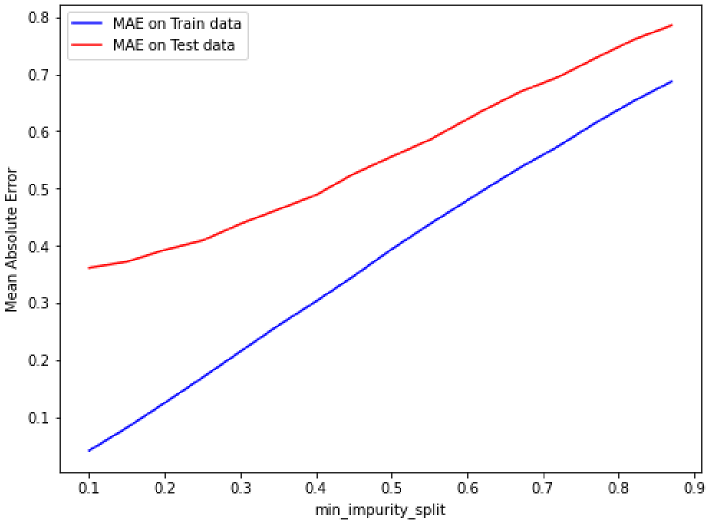


Figure 6.30: MAE scores of the Decision Tree model on *min_impurity_split*

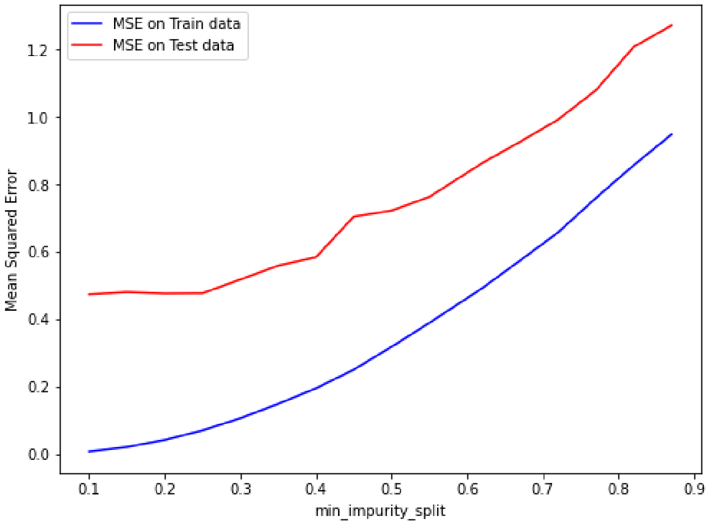


Figure 6.31: MSE scores of the Decision Tree model on *min_impurity_split*

In Figure 6.30 and Figure 6.31, we observe that there is overfitting for lower values of *min_impurity_split*. The optimal performance is achieved at $min\_impurity\_split = 0.8$. We observe that as we increase the *min_impurity_split*, the error values keep increasing, however, after 0.75, the difference between the train and test errors remains consistent. There is a significant change in the errors and δE values. The performance is satisfactory and can be compared with other methods for evaluation.

6.7.2 Random Forest (RF) Regression

Random Forest is an ensemble regression technique that uses bagging technique. We briefly discussed the random forest regression in the previous chapter. We used the SciKit-learn [47] library for the implementation of this algorithm. The design choices of the algorithm includes selecting *n_estimators* or the number of trees in the forest. The other parameters are set to default and need to be changed in case of significant overfitting. The *criterion* is set to *MAE* instead of *MSE*. It was observed that the *good models* have very low MSE, which made them difficult to compare with other models with similar values. However, this was not the case with MAE. Hence, MAE is always the choice of the loss function and the primary criteria to assess the quality of the models with this dataset.

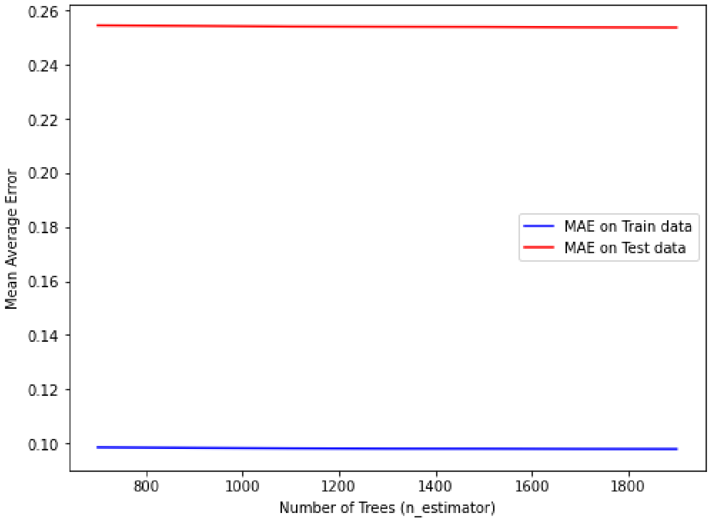


Figure 6.32: MAE scores of the Random Forest model on $n\_estimators$

In Figure 6.32, there is a little overfitting, However, this is not very significant and suggests that the number of trees for the subsequent algorithms can remain default (of 100). The other metrices E.3 such as MSE, δE_{avg}, δE_{95} and δE_{max} also have similar trend.

6.7.3 Extra Tree (ET) Regression

As we discussed in Section 5.10.2, the basic principle of extra trees is similar to that of random forest. However, unlike random forest, the extra trees use a random subset of features to train each base estimator [37]. We used the SciKit-learn [47] library of the extra trees for implementation and the results are as follows.

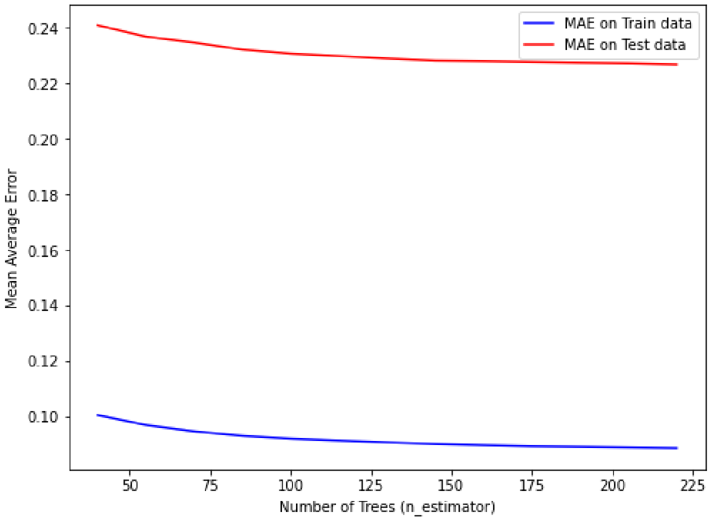


Figure 6.33: MAE scores of the Extra Trees Tree model on $n\_estimators$

In Figure 6.33, we observe that there is a marginal (but tolerable) overfitting. Other metrices also exhibit similar trends.

6.7.4 Adaboost (AB) Regression

In Section 5.10.2 we saw that the Adaptive Boost regression uses weak learners that improve the prediction error over iterations. The SciKit learn [47] library was used for the implementation. The model was iterated over the number of trees and the learning rates. The results obtained are in Figures 6.34 and 6.35.

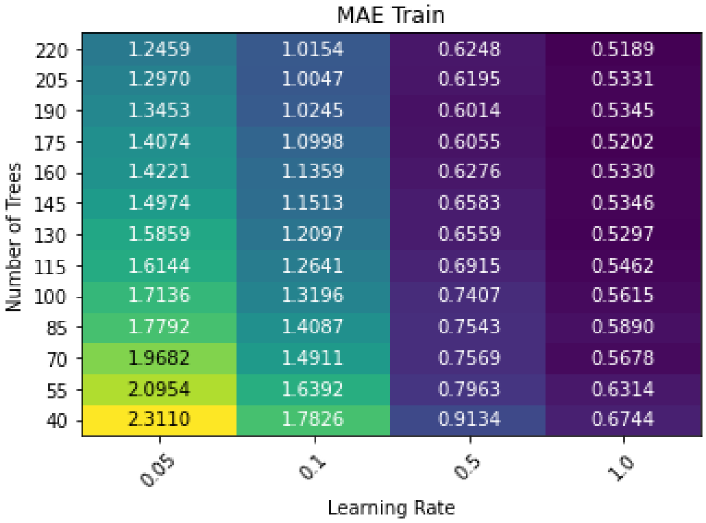


Figure 6.34: Heatmap of MAE scores of the Adaboost models on the training data

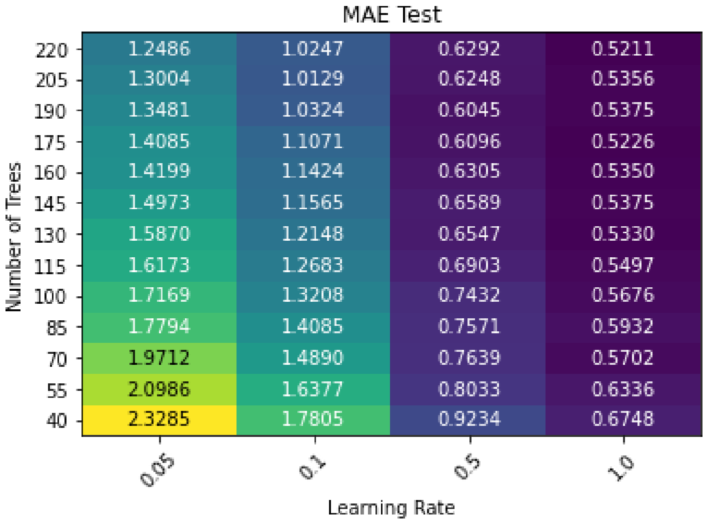


Figure 6.35: Heatmap of MAE scores of the Adaboost models on the testing data

We observe that the MSE and MAE are in the similar scale. The MAE decreases with the increase in learning rate and increase in the number of trees. The boosted ensemble is made from the *DecisionTreeRegressor* as the errors of the Decision Tree regressor are low. Hence, the cumulative result of such trees performs well. The loss function used for all the tested models was *squared* as the model had lower error values and lesser δE values compared to other available loss functions like *linear* and *exponential*.

6.7.5 Gradient Boosting (GB) Regression

In Section 5.10.2, we discussed on how the Gradient Boost identifies the shortcomings by using gradients in the loss function instead of using high weight data points in AdaBoost. Figure 6.36 and Figure 6.37 show training and testing performance of the model on our dataset.

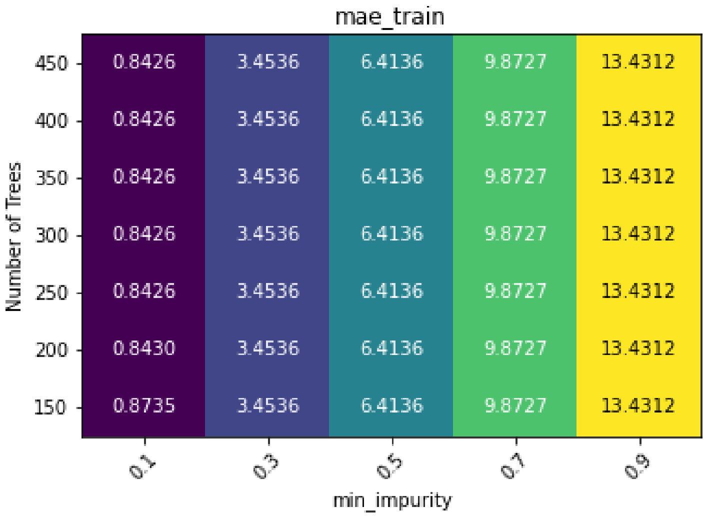


Figure 6.36: Heatmap of MAE scores of Gradient Boost models on the training data

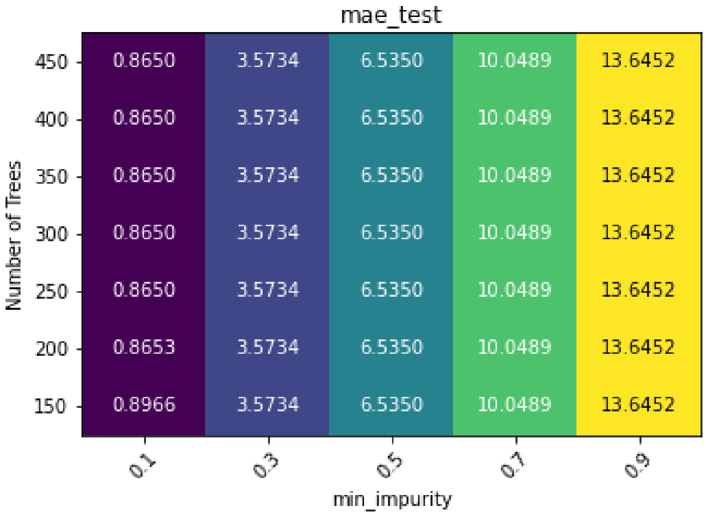


Figure 6.37: Heatmap of MAE scores of Gradient Boost models on the testing data

We observe that that the number of trees do not really have any significant impact (after a point) on the performance. Moreover, high is the impurity, higher is the error. This can be attributed to the fac that if the final impurity decrease is less than the minimum impurity decrease, then the split will not be performed. As this threshold kept increasing, the trees started becoming shorter and shorter, deteriorating the results.

6.7.6 Extreme Gradient Boosting (XGB) Regression

We discussed the XGBoost regression algorithm in Section 5.10.2. The key difference between the Extreme Gradient Boosting (XGBoost) algorithm and the regular Gradient Boosting algorithm is that the regular Gradient Boosting uses the loss function of the base model (e.g. decision trees) to minimize the error of the overall model, XGBoost uses the 2nd order derivative for approximation. Unlike the Gradient Boosting the XGBoost uses L1 and L2 regularization techniques to reduce overfitting and improve model generalization. We used the python library XGBoost [10] for implementation. The results are as follows (refer to Figure 6.38 and Figure 6.39).

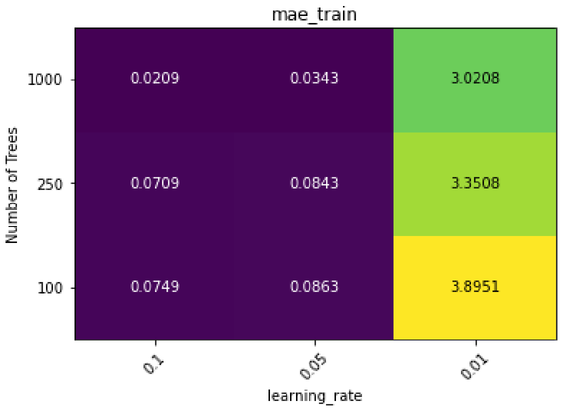


Figure 6.38: Heatmap of MAE scores of XGBoost models on the training data

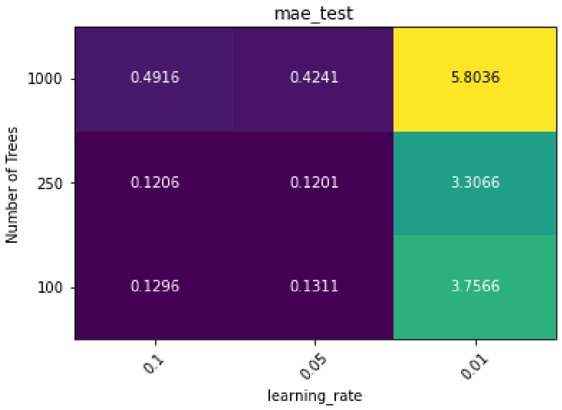


Figure 6.39: Heatmap of MAE scores of XGBoost models on the testing data

In Figure 6.38 and Figure 6.39, we observe that the MAE for XGBoost increases with decreasing learning rate and abruptly high at 0.01. Moreover, the error decreases with increasing number of trees in the forest. However, for when there are 1000 trees in the forest, it leads to overfitting. Hence, the 250 trees with 0.05 learning rate provides the optimal result. Moreover, the MSE and MAE scores are the lowest we have observed so far with other algorithms. Other metrices (such as the MSE, δE_{avg}, δE_{95} and δE_{max})also follow the similar trend. The details are provided in the Appendix E.7.

6.8 Support Vector (SVR) Regression

The Support Vector Regression (SVR) is based on the same basic principles as the SVM (for classification). However, there are minor differences. In SVR, we have a decision boundary from the original hyperplane such that the Support Vectors are within the decision boundaries. The algorithm is briefly explained in Section 5.11. The SciKit-learn [47] library was used for the implementation of this model.

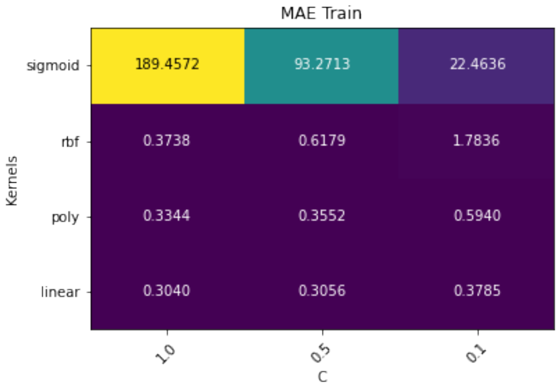


Figure 6.40: MAE scores of the SVR on train data

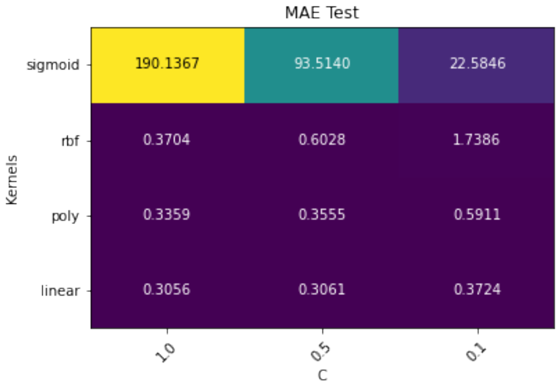


Figure 6.41: MAE scores of the SVR on test data

In Figure 6.40 and Figure 6.41, we observe that the linear kernel has the least error, followed by Polynomial, RBF and Sigmoid. Further, as we increase the value of the regularization parameter, the value of the error increases. Similar trends are followed by other metrices as well (Appendix E.8).

6.9 Comparative Analysis

Table 6.4 represents *best* model (in terms of lowest δE_{95}) from each of the algorithms. It should be noted that the data set used for the DNN network is different compared to the other methods to obtain the following results.

	MLP	DNN	DT	RF	ET	GB	AB	XGB	SV
MAE	0.279	0.294	0.361	0.255	0.512	0.381	0.522	0.120	0.306
MSE	0.144	0.193	0.473	0.144	0.474	0.417	0.447	0.028	0.164
95% δE	0.712	0.704	1.320	0.781	1.504	1.248	1.916	0.392	1.175
Avg δE	0.340	0.374	0.461	0.323	0.666	0.491	1.029	0.173	0.558
Max δE	2.248	2.813	13.50	4.251	2.816	9.704	3.313	1.604	4.536

Table 6.4: Results from all the regression algorithms

Let's begin the analysis with the *best* model. Here, the best model is the model that has the minimum δE_{95}. We observe that the least MSE, MAE and δE_{95} is achieved with XGBoost. In fact, this is the only algorithm that has δE_{95} lower than the threshold of 0.5 (we set in the beginning). The trend of MSE and MAE is very similar to that of δE_{95}. Moreover, if we compare the performance of the MLP model and the deep neural network, we observe that even though the deep neural network is more complex and uses 2 extra features (Region and Coating), the difference between the respective errors is not very significant. However, these results are not the best representatives of what we can expect when a new medium is tested with the model.

6.10 Validation

Table 6.5 illustrates the performance of the XGBoost algorithm when the indicated medium was removed from the training data set and used as an "unknown medium" to the model for testing. Each medium is randomly selected from each of the 5 suppliers.

Media	MAE	MSE	δE_{95}	δE_{avg}	δE_{max}
GPrint Arctic Paper 115gsm	0.361	0.226	0.73	0.44	1.37
UPM Digi Finesse Premium Silk 115gsm	0.485	0.418	0.82	0.55	2.36
Sappi Flo Digital Gloss Text 118gsm	0.280	0.138	0.65	0.34	1.80
Verso Sterling Premium Silk Cover 271gsm	0.284	0.160	0.69	0.32	1.12
Sappi Magno Plus Silk 115gsm	0.284	0.166	0.71	0.35	1.58

Table 6.5: Performance of XGBoost with different media

The δE_{95} with these results are higher than 0.5 in all these cases. This is a deviation of results from the test dataset. However, the results suggest that the errors and δE are consistent across all different suppliers with a bit higher error for UPM supplier. The reason could be higher drying effect of the media by UPM or relatively smaller data set for training or a combination of both.

To investigate the first reasoning, we conduct a test. We find the correlation between δE_X_y and $\delta E\_y\_pred\_y$. Please refer to Figure 2.2 for visualization. However, this time instead of X_test and y_test, we use the entire data set (hence, X and y) to make predictions and compute δE. The Pearson correlation between δE_X_y and $\delta E\_y\_pred\_y$ is 0.28. This suggests that there is not a very significant correlation between the actual drying effect and the error in prediction. Hence, we can conclude that the higher relative δE is attributed to smaller data set. We had used 17 different media from 5 suppliers. The training data set is decided based on the suggestions by the experts and stock in the warehouses. Hence, it can be concluded that the data set might not be the best representative of all the coated media in the industry. Moreover, the unavailability of dry back data for the media posed a challenge while selecting the data set for training the models.

When the validation is performed on the validation data set (Refer to Table 5.4), it is observed that the model does not perform very well for media that belong to a different region. Table 6.6 shows the actual drying for each of the media used in the validation.

Media	δE_{95}	δE_{avg}	δE_{max}
Gold East Space Shuttle 157gsm	3.15	1.80	8.1
Zanders Silver Digital 200gsm	1.60	1.00	2.24
Zhonghua Ninbo Star 250 gsm	3.96	2.10	9.54

Table 6.6: Actual Drying Pattern of Validation Data

Table 6.7 shows the δE between the $L*$, $a*$, $b*$ values at Time 96 and the predicted $L*$, $a*$, $b*$ values.

Media	**MAE**	**MSE**	δE_{95}	δE_{avg}	δE_{max}
Gold East Space Shuttle 157gsm	1.00	1.98	2.65	1.20	7.41
Zanders Silver Digital 200gsm	0.32	0.18	0.72	0.38	1.63
Zhonghua Ninbo Star 250gsm	1.15	3.21	3.76	1.43	9.53
Sappi Magno Plus 150gsm	0.501	0.439	0.85	0.49	1.76

Table 6.7: Errors and δE in prediction of drying data

There are 2 media from China that have significantly high actual δE. Perhaps, there are some factors that the model has not learned from the given data set. However, the predicted δE_{95} is relatively low for *Zanders' Silver Digital* (from EU) for decently high actual δE. Hence, it can be concluded that the data set considered for training is not representative of all the coated media. Hence, the drying effect of media from different region needs to be studied before they are used for prediction in the model.

6.11 Summary

In this chapter, we studied the results and obtained key observations about the data, the regression techniques and the predicted outputs. We evaluated each of the models under different hyper-parameters and compared them with each other. In the next chapter, we will draw final conclusions from the results and discuss come other salient observations that are not directly related to this study but can be used for research pertaining to dry back predictions. Furthermore, we discuss the challenges and future scope of this project.

Chapter 7

Conclusion

In this chapter, we conclude this project. Section 7.1 summarizes the project. The challenges and future scope of the project is covered in Section 7.2. Further, we discuss some of the miscellaneous tests we performed during the course of the project but are not directly related to our research problem.

7.1 Concluding Summary

In the beginning of this project, we set an aim to study the effects of the factors that contribute to the dry back. We limited the scope of our project to cover only the media aspect of it. Based on these factors, we then make a regression model to predict the $L*$, $a*$ and $b*$ values after the color has stabilized. We can divide the data set into 3 parts, the colors (i.e. $CMYK$), the $L*a*b*$ and the media characteristics including the region, coating and supplier. During the implementation and testing of the models, it was observed that there is a disparity between the importance of each of these factors on the dry back. The $L*$, $a*$ and $b*$ values at Time 96 depend on the $L*$, $a*$ and $b*$ at Time 00 while the other factors only have a little contribution.

This behavior is expected because the new $L*$, $a*$ and $b*$ are the just the old $L*$, $a*$ and $b*$ values at a later point of time. As far as the other factors are concerned, the color also plays an important role. For instance, the higher quantity of yellow color showed higher δE and longer drying time. Experts suggest that this is a known phenomenon and this had been demonstrated in the CPP in the past. The media characteristics have significantly lower impact on the dry back but when compared among themselves, weight and gloss seem to have high importance.

Experts have suggested that the dry back is a surface phenomenon and its extent depends on the size of the pores at the surface of the medium (along with other factors) as the pores absorb the ink. The coating at the surface affects the size of the pores. This would explain why the gloss percentage has importance in predicting the dry back. If this reasoning is true, there should have been other characteristics that would have had similar importance scores such as the brightness. However, the reason we did not find this could be because of the lack of data of brightness.

In a study conducted before the data collection had revealed that the weight does not have a very significant role in determining the dry back. This conclusion is backed by the evidence that there was no pattern whatsoever between the δE and the weight when all the other factors remained constant. However, during the prediction it is observed that the importance is comparable to the media characteristics adding the weight number only improved the models. Each medium has only a maximum of two weight categories. This ensured a balance and helped us cover more media and suppliers.

The results from Section 6.9 suggest that the best algorithm to use is the XGBoost as it provides the least δE. Since the model is supposed to predict the dry back for new (which is unknown) medium, we simulated the scenario by taking a medium out of the training dataset and retraining the model. Now, the medium was used to make predictions and hence, evaluate the model. The results suggest that the δE_{95} is not very consistent across all the media. When the aforementioned exercise was performed with some media, the δE_{95} varied from 0.4 to 0.8. However, the color experts have suggested that this error is within the tolerable limit.

The results from validation (Section 6.10) suggest that this trained model is not good enough to be used for all the coated media. Since, it shows relatively good performance in case of EU and US media, the model can be used for media from these regions. The 2 media used for validation from China cannot be used as a conclusive evidence to suggest that the Asian media have high dry back. This means that the model can still be used on media from other regions that do not have very high dry back. Since it can be tricky to determine if a new (coated) media from a different region has very high dry back or not without actually finding out the δE, the media properties can be used to compare it with the current data on media characteristics to have an estimation of dry back and decide whether to use the model.

7.2 Limitations and Future Scope

The biggest challenge in this project was around the data set. The data set contains 17 *different types* of media from 5 suppliers. This is minuscule compared to the actual numbers in the global market. Hence, it cannot be guaranteed that the dataset is representative of all the coated media. The results are satisfactory in the sense that the error (δE) is low and the (predicted) color profile is similar to that of the actual dried one in case of the American and European coated media. However, the validation results highlight the need to have a bigger training dataset to enhance the generalizability. The challenge around the media characteristics has been discussed in the implementation section 5.3.2. The deficiency of data of the media characteristics has restricted their comprehensive study and use in the regression models.

A major challenge was around the availability of data on dry back. The phenomena of dry back has not been studied thoroughly in the context of media in the past and very limited information is available in the public domain. For instance, there is no information available on the typical δE we can expect on dry back of coated media for different suppliers (and coatings). As a result, we are not able to determine if the current data set (used for training) is representative of the drying behavior. The validation results reveal that it is not representative. However, we are not able to determine the actual range of δE (for dry back) across different coated media. Experts suggest that the dry back is dependent upon the composition of materials used in the paper and it is known that the composition varies by region. However, the extent of its influence is yet to be determined (and quantified). The entire process of data collection is relatively lengthy and demands availability of various different types of media during this process.

The data set for this project was obtained from the Niagara R4 engine. Hence, the validity of the model can be ensured only for the printers that use the R4 engine. Since, the print conditions change with the engine version, its effects on the dry back (and hence, the predictions) are yet to be determined. For instance, the R2 engine does not have the super-heated steam which reduces the dry back effect. Hence, the data set obtained from R2 engine is expected to have relatively higher δE for a given medium. However, the configuration of the model would remain the same.

Furthermore, there are a plethora of printer settings that may or may not have an impact on drying. Since all the printer settings were maintained as constants in our study, if the regression model needs to be used in the printer software, it needs to be tested in all the conditions. The printer settings that do affect the drying patterns should be studied and factored into the model (perhaps by adding more features). The printer settings for each of the prints (by the printer) are logged in the media catalog. The media catalogs of the printers would be good starting points to collect the data on the printer settings.

In the future, a more diverse training data set would help in better training of the model. Even though, the δE and the errors in the test data set might still not improve, such a model will be able to accommodate new suppliers from different regions of the world and enhance generalizability. The feature importance test on the media characteristics revealed that brightness has relatively higher contribution in the predictions. Since we are dependent upon the suppliers to provide the data on media characteristics, if we are able to collect the data ourselves (perhaps by installing sensors in the printers), it will ensure consistent and reliable data which can consequently improve the accuracy of the model.

Bibliography

[1] Muhammad Waseem Ahmad, Jonathan Reynolds, and Yacine Rezgui. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of Cleaner Production*, 203:810–821, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0959652618325551, doi:https://doi.org/10.1016/j.jclepro.2018.08.207. 16

[2] Yenew Alemu. LAP LAMBERT Academic Publishing, 2019. 8

[3] T. W. Anderson and D. A. Darling. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193 – 212, 1952. doi:10.1214/aoms/1177729437. 8

[4] Dibya Bora. Importance of image enhancement techniques in color image segmentation: A comprehensive and comparative study. *Indian Journal of Scientific Research*, 15:115–131, 07 2017. doi:10.6084/m9.figshare.5280799. 5

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144152, New York, NY, USA, 1992. Association for Computing Machinery. doi:10.1145/130385.130401. 18

[6] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. URL: http://link.springer.com/10.1007/BF00058655, doi:10.1007/BF00058655. 16

[7] Patricio Cerda, Gal Varoquaux, and Balzs Kgl. Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107(8-10):14771494, 2018. doi:10.1007/s10994-018-5724-2. 23

[8] B. Chandrasekaran and A.K. Jain. Quantization complexity and independent measurements. *IEEE Transactions on Computers*, C-23(1):102–106, 1974. doi:10.1109/T-C.1974.223789. 24

[9] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. Cumulative attribute space for age and crowd density estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2013. doi:10.1109/CVPR.2013.319. 22

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785794, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2939672.2939785. 17, 29, 30, 50

[11] Tianqi Chen, Sameer Singh, Ben Taskar, and Carlos Guestrin. Efficient second-order gradient boosting for conditional random fields. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015. URL: http://terraswarm.org/pubs/481.html. 17

[12] Francois Chollet et al. Keras, 2015. URL: https://github.com/fchollet/keras. 42, 44

[13] Stephen P. Curram and John Mingers. Neural networks, decision tree induction and discriminant analysis: an empirical comparison. *Journal of the Operational Research Society*, 45(4):440–450, 1994. doi:10.1057/jors.1994.62. 15

[14] B.V. Dasarathy and B.V. Sheela. A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, 67(5):708–713, 1979. doi:10.1109/PROC.1979.11321. 15

[15] R. Doornbos. *Testing for an outlier in a linear model*. Memorandum COSOR. Technische Hogeschool Eindhoven, 1980. 23

[16] Jane Elith*, Catherine H. Graham*, Robert P. Anderson, Miroslav Dudk, Simon Ferrier, Antoine Guisan, Robert J. Hijmans, Falk Huettmann, John R. Leathwick, Anthony Lehmann, Jin Li, Lucia G. Lohmann, Bette A. Loiselle, Glenn Manion, Craig Moritz, Miguel Nakamura, Yoshinori Nakazawa, Jacob McC. M. Overton, A. Townsend Peterson, Steven J. Phillips, Karen Richardson, Ricardo Scachetti-Pereira, Robert E. Schapire, Jorge Sobern, Stephen Williams, Mary S. Wisz, and Niklaus E. Zimmermann. Novel methods improve prediction of species distributions from occurrence data. *Ecography*, 29(2):129–151, 2006. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2006.0906-7590.04596.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2006.0906-7590.04596.x, doi:https://doi.org/10.1111/j.2006.0906-7590.04596.x. 17

[17] Junliang Fan, Xiukang Wang, Lifeng Wu, Hanmi Zhou, Fucang Zhang, Xiang Yu, Xianghui Lu, and Youzhen Xiang. Comparison of support vector machine and extreme gradient boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in china. *Energy Conversion and Management*, 164:102–111, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0196890418302048, doi:https://doi.org/10.1016/j.enconman.2018.02.087. 30

[18] Yu Feng, Ningbo Cui, Qingwen Zhang, Lu Zhao, and Daozhi Gong. Comparison of artificial intelligence and empirical models for estimation of daily diffuse solar radiation in north china plain. *International Journal of Hydrogen Energy*, 42(21):14418–14428, 2017. URL: https://www.sciencedirect.com/science/article/pii/S0360319917314738, doi:https://doi.org/10.1016/j.ijhydene.2017.04.084. 28, 30

[19] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 148156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. 17

[20] G. Fumera and F. Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, 2005. doi:10.1109/TPAMI.2005.109. 16

[21] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006. URL: http://link.springer.com/10.1007/s10994-006-6226-1, doi:10.1007/s10994-006-6226-1. 16

[22] Frank E. Grubbs. Sample Criteria for Testing Outlying Observations. *The Annals of Mathematical Statistics*, 21(1):27 – 58, 1950. doi:10.1214/aoms/1177729885. 23

[23] Isabelle Guyon, Bernhard E. Boser, and Vladimir Vapnik. Automatic capacity tuning of very large vc-dimension classifiers. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, page 147155, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. 18

[24] Muhammed A. Hassan, A. Khalil, S. Kaseb, and M.A. Kassem. Exploring the potential of tree-based ensemble methods in solar radiation modeling. *Applied Energy*, 203:897–916, 2017. URL: https://www.sciencedirect.com/science/article/pii/S0306261917308437, doi:https://doi.org/10.1016/j.apenergy.2017.06.104. 28, 30

[25] Csaba Horvath and Pl Grgnyi-Tth. Study of colour change in the course of drying on prints created using offset printing technology. *Proceedings of 9th International Symposium on Graphic Engineering and Design*, 2018. doi:10.24867/grid-2018-p39. 2, 3, 4

[26] ISO. Paper and board determination of smoothness (bekk method). ISO 5627:1995, International Organization for Standardization, Geneva, Switzerland, 1995. 14

[27] ISO. Paper and board determination of cie whiteness, d65/10 degrees (outdoor daylight). ISO 11475:2004, International Organization for Standardization, Geneva, Switzerland, 2004. 13, 14

[28] ISO. Paper and board determination of opacity (paper backing) diffuse reflectance method. ISO 2471:2008, International Organization for Standardization, Geneva, Switzerland, 2008. 13

[29] ISO. Paper and board measurement of specular gloss part 1: 75 degree gloss with a converging beam, tappi method. ISO 8254-1:2009, International Organization for Standardization, Geneva, Switzerland, 2009. 13

[30] ISO. Paper and board determination of thickness, density and specific volume. ISO 534:2011, International Organization for Standardization, Geneva, Switzerland, 2011. 13, 14

[31] ISO. Graphic technology - process control for the production of half-tone colour separations, proof and production prints - part 2: Offset lithographic processes. ISO 12647-2:2013, International Organization for Standardization, Geneva, Switzerland, 2013. 7

[32] ISO. Paper and board measurement of specular gloss part 2: 75 degree gloss with a parallel beam, din method. ISO 8254-2:2016, International Organization for Standardization, Geneva, Switzerland, 2016. 13

[33] ISO. Paper and board measurement of specular gloss part 3: 20 degree gloss with a converging beam, tappi method. ISO 8254-3:2016, International Organization for Standardization, Geneva, Switzerland, 2016. 13

[34] ISO. Paper, board and pulps measurement of diffuse blue reflectance factor part 1: Indoor daylight conditions (iso brightness). ISO 2470-1:2016, International Organization for Standardization, Geneva, Switzerland, 2016. 14

[35] ISO. Paper, board and pulps measurement of diffuse blue reflectance factor part 2: Outdoor daylight conditions (d65 brightness). ISO 2470-2:2016, International Organization for Standardization, Geneva, Switzerland, 2016. 14

[36] Jitendra Kumar Jaiswal and Rita Samikannu. Application of random forest algorithm on feature subset selection and classification and regression. In *2017 World Congress on Computing and Communication Technologies (WCCCT)*, pages 65–68, 2017. doi:10.1109/WCCCT.2016.25. 16

[37] Vijay John, Zheng Liu, Chunzhao Guo, Seiichi Mita, and Kiyosumi Kidono. Real-time lane estimation using deep features and extra trees regression. In Thomas Bräunl, Brendan McCane, Mariano Rivera, and Xinguo Yu, editors, *Image and Video Technology*, pages 721–733, Cham, 2016. Springer International Publishing. 16, 47

[38] Frank J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769, doi:10.1080/01621459.1951.10500769. 33

[39] Deane B. Judd, David L. MacAdam, Günter Wyszecki, H. W. Budde, H. R. Condit, S. T. Henderson, and J. L. Simonds. Spectral distribution of typical daylight as a function of correlated color temperature. *J. Opt. Soc. Am.*, 54(8):1031–1040, Aug 1964. URL: http://www.osapublishing.org/abstract.cfm?URI=josa-54-8-1031, doi:10.1364/JOSA.54.001031. 21

[40] Manoj Kuppusamy and Senthamarai Kaliyaperumal. Comparison of methods for detecting outliers. *International Journal of Scientific & Engineering Research*, 4:709–714, 01 2013. 23

[41] J. R. Leathwick, J. Elith, M. P. Francis, T. Hastie, and P. Taylor. Variation in demersal fish species richness in the oceans surrounding new zealand: an analysis using boosted regression trees. *Marine Ecology Progress Series*, 321:267–281, 2006. URL: https://doi.org/10.3354/meps321267. 17

[42] RODERICK J. A. LITTLE and DONALD B. RUBIN. The analysis of social science data with missing values. *Sociological Methods & Research*, 18(2-3):292–326, 1989. arXiv:https://doi.org/10.1177/0049124189018002004, doi:10.1177/0049124189018002004. 21

[43] Frank J. Massey. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78, March 1951. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769, doi:10.1080/01621459.1951.10500769. 64

[44] R. Mcdonald and K J Smith. Cie94-a new colour-difference formula*. *Journal of the Society of Dyers and Colourists*, 111(12):376379, 2008. doi:10.1111/j.1478-4408.1995.tb01688.x. 5

[45] James N. Morgan and John A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):415–434, 1963. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500855, doi:10.1080/01621459.1963.10500855. 15

[46] Sokratis Papadopoulos, Elie Azar, Wei-Lee Woon, and Constantine E. Kontokosta. Evaluation of tree-based ensemble learning algorithms for building energy performanceestimation. *Journal of Building Performance Simulation*, 11(3):322–332, 2018. doi:10.1080/19401493.2017.1354919. 28, 30

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 16, 23, 24, 25, 26, 28, 29, 30, 38, 40, 45, 46, 47, 48, 51

[48] Sebastian Raschka. Feature importance permutation. URL: http://rasbt.github.io/mlxtend/user_guide/evaluate/feature_importance_permutation/. 40, 41, 66

[49] Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to pythons scientific computing stack. *The Journal of Open Source Software*, 3(24), April 2018. URL: http://joss.theoj.org/papers/10.21105/joss.00638, doi:10.21105/joss.00638. 26

[50] Poornachandra Sarang. Neural networks for regression. *Artificial Neural Networks with TensorFlow 2*, page 189230, 2020. doi:10.1007/978-1-4842-6150-7_5. 23, 24

[51] Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2):197227, July 1990. doi:10.1023/A:1022648800760. 17

[52] Jun Shao and Randy R. Sitter. Bootstrap for imputed survey data. *Journal of the American Statistical Association*, 91(435):1278–1288, 1996. URL: https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1996.10476997, doi:10.1080/01621459.1996.10476997. 21

[53] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/col.20070, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.20070, doi:https://doi.org/10.1002/col.20070. 5, 6

[54] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Aug 2004. doi:10.1023/B:STCO.0000035301.49549.88. 18

[55] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9:307, 08 2008. doi:10.1186/1471-2105-9-307. 40, 66

[56] Qing Tian, Hui Xue, and Lishan Qiao. Human age estimation by considering both the ordinality and similarity of ages. *Neural Processing Letters*, 43(2):505521, 2015. doi:10.1007/s11063-015-9423-8. 22

[57] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):306–307, 1979. doi:10.1109/TPAMI.1979.4766926. 24

[58] Geoffrey K.F. Tso and Kelvin K.W. Yau. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9):1761–1768, 2007. URL: https://www.sciencedirect.com/science/article/pii/S0360544206003288, doi:https://doi.org/10.1016/j.energy.2006.11.010. 15

[59] John W. Tukey. *Exploratory data analysis*. Pearson, 2020. 23

[60] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):32213245, January 2014. 25

[61] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008. 25

[62] Laurens van der Maaten and Geoffrey Hinton. Visualizing non-metric similarities in multiple maps. *International Journal of Advanced Manufacturing Technology - INT J ADV MANUF TECHNOL*, 87, 01 2011. doi:10.1007/s10994-011-5273-4. 25

[63] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi:10.25080/Majora-92bf1922-00a. 66

Appendix A

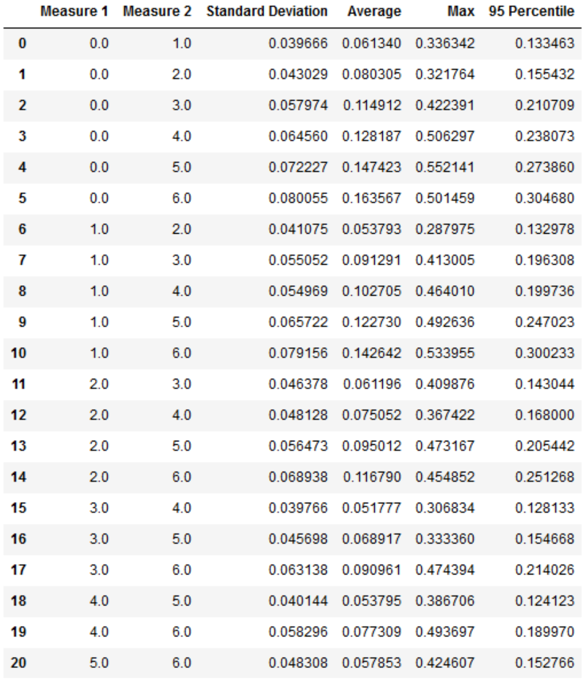
Terms and Definitions

- **Bootstrapping:** Bootstrap is a method of resampling which is used to estimate the statistics on a population by sampling a dataset with replacement.
- **CYMK (Cyan Yellow Magenta Black):** These are the primary colors used in a printer. All the other colors can be derived from combination of these colors. The representation of these colors is done on percentage. Hence, the value of each constituent color is from 0 to 100.
- **Dry back:** It is a phenomena of decrease in the density and gloss of the wet freshly printed ink film upon drying.
- **Media:** It is a physical substrate that can be processed by a printer. It is a tangible object or can be laid down somewhere.
- **TAC (Total Area Coverage):** Also known as Ink Density or Total Ink Coverage, TAC is the limit of the amount of ink that can be put on the medium. The unit is ml/m^2
- **Range:** The difference between maximum and minimum values in the given category.

Appendix B

Measurement Device Accuracy Test

Table B.1 represents the standard deviation, average, maximum and 95 percentile of the δE between each pair of measurement. The measurements are index from 0 to 6.



	Measure 1	Measure 2	Standard Deviation	Average	Max	95 Percentile
0	0.0	1.0	0.039666	0.061340	0.336342	0.133463
1	0.0	2.0	0.043029	0.080305	0.321764	0.155432
2	0.0	3.0	0.057974	0.114912	0.422391	0.210709
3	0.0	4.0	0.064560	0.128187	0.506297	0.238073
4	0.0	5.0	0.072227	0.147423	0.552141	0.273860
5	0.0	6.0	0.080055	0.163567	0.501459	0.304680
6	1.0	2.0	0.041075	0.053793	0.287975	0.132978
7	1.0	3.0	0.055052	0.091291	0.413005	0.196308
8	1.0	4.0	0.054969	0.102705	0.464010	0.199736
9	1.0	5.0	0.065722	0.122730	0.492636	0.247023
10	1.0	6.0	0.079156	0.142642	0.533955	0.300233
11	2.0	3.0	0.046378	0.061196	0.409876	0.143044
12	2.0	4.0	0.048128	0.075052	0.367422	0.168000
13	2.0	5.0	0.056473	0.095012	0.473167	0.205442
14	2.0	6.0	0.068938	0.116790	0.454852	0.251268
15	3.0	4.0	0.039766	0.051777	0.306834	0.128133
16	3.0	5.0	0.045698	0.068917	0.333360	0.154668
17	3.0	6.0	0.063138	0.090961	0.474394	0.214026
18	4.0	5.0	0.040144	0.053795	0.386706	0.124123
19	4.0	6.0	0.058296	0.077309	0.493697	0.189970
20	5.0	6.0	0.048308	0.057853	0.424607	0.152766

Figure B.1: Device Accuracy Test

Appendix C

Kolmogorov-Smirnov Test on Device Accuracy Test Data

When the Kolmogorov-Smirnov Test [43] was performed on each pair of δE (Figure B.1) to investigate if the δE from the same device and same test chart belong to the same distribution.

C.0.1 Setup

The δE values computed in the accuracy test (Appendix B) were used in this study. The Kolmogorov-Smirnov Test [43] was performed on each of the pairs of these δE distributions.

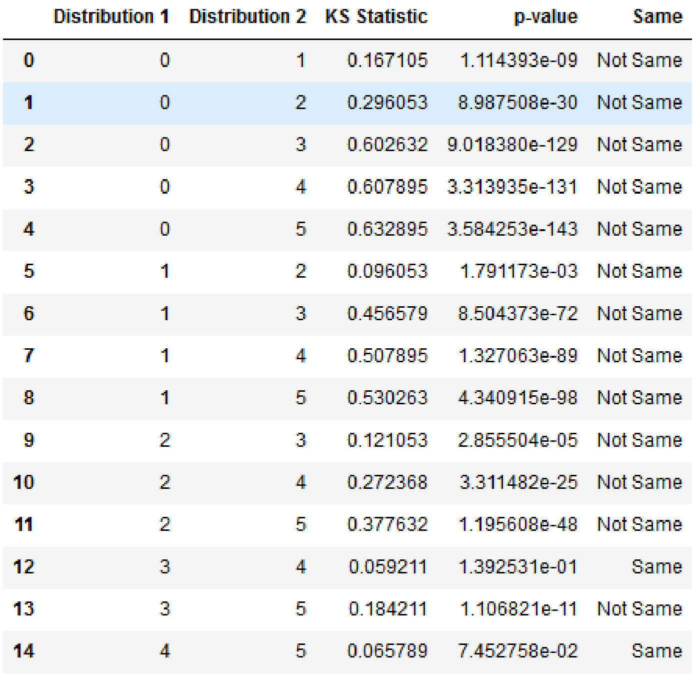
C.0.2 Results

Each δE distribution is indexed in the order of their measurement. Table C.1 represents the test statistic of the Kolmogorov-Smirnov Test [43], its p-value and conclusion whether the distributions are same based on the p-value.

C.0.3 Conclusion

The result suggests that the most of the δE distributions are **not** the same. However, it is interesting to note that these distributions are coming from the same sheet and measured with the same measuring device. Ideally, the values of each δE should be 0. Based on the results of the Device Accuracy Test (Appendix B), it was expected that these sets of δE would belong to the same distribution.

Hence, it can be concluded that the measurement error is significant enough to change the distribution of δE over a number of iterations.



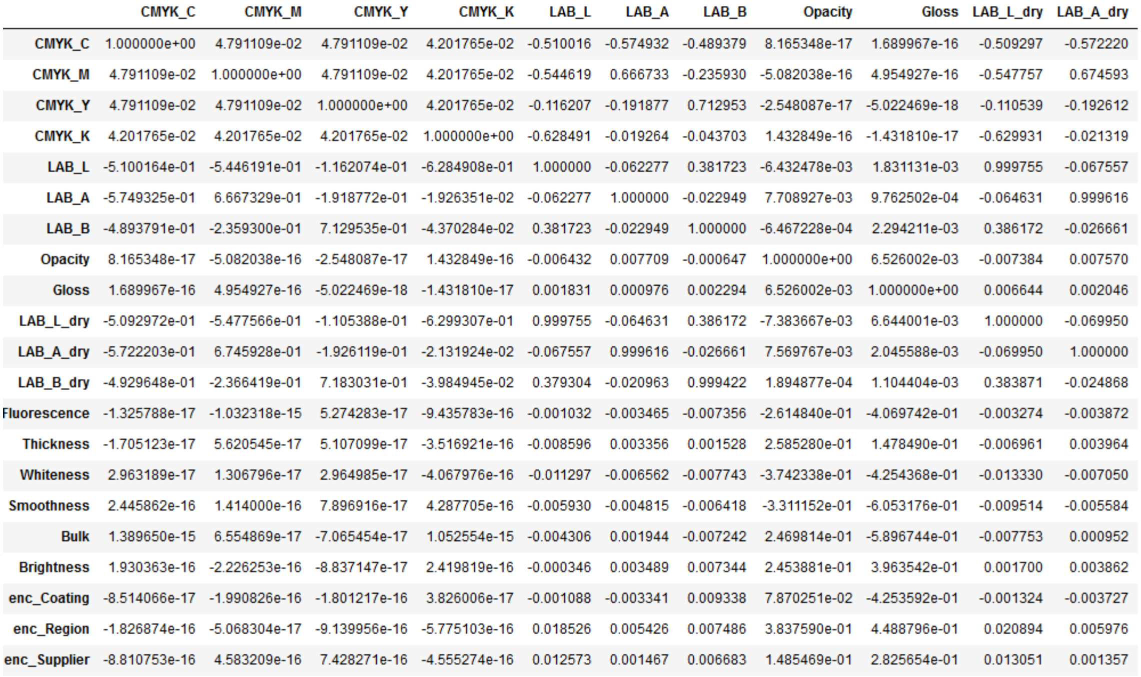
	Distribution 1	Distribution 2	KS Statistic	p-value	Same
0	0	1	0.167105	1.114393e-09	Not Same
1	0	2	0.296053	8.987508e-30	Not Same
2	0	3	0.602632	9.018380e-129	Not Same
3	0	4	0.607895	3.313935e-131	Not Same
4	0	5	0.632895	3.584253e-143	Not Same
5	1	2	0.096053	1.791173e-03	Not Same
6	1	3	0.456579	8.504373e-72	Not Same
7	1	4	0.507895	1.327063e-89	Not Same
8	1	5	0.530263	4.340915e-98	Not Same
9	2	3	0.121053	2.855504e-05	Not Same
10	2	4	0.272368	3.311482e-25	Not Same
11	2	5	0.377632	1.195608e-48	Not Same
12	3	4	0.059211	1.392531e-01	Same
13	3	5	0.184211	1.106821e-11	Not Same
14	4	5	0.065789	7.452758e-02	Same

Figure C.1: Kolmogorov-Smirnov Test on the Accuracy Test Dataset

Appendix D

Correlation Between Variables

Some algorithms such as the feature importance [48] are prone to biased results if the predicted quantities are highly correlated to the input features [55]. Moreover, some of these interesting results can be used in other projects at the organization as well. We performed an analysis to study the correlation between all the variables. We used the Pandas' [63] implementation for Pearson's correlation and the results are as follows in Figure D.1.

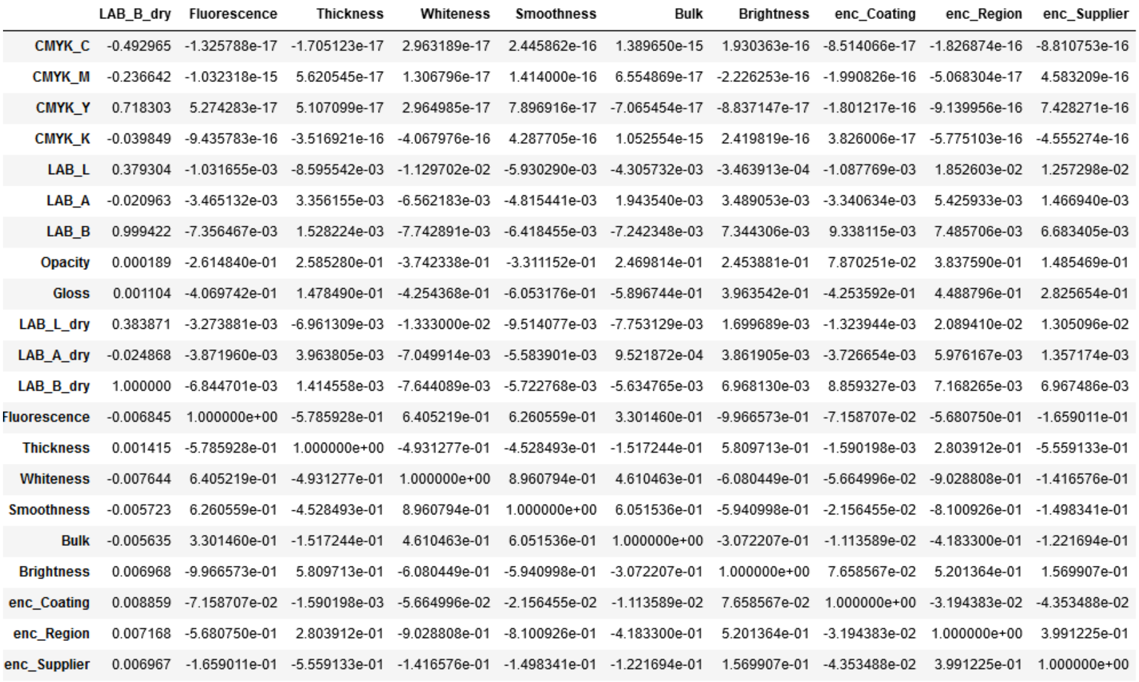


	CMYK_C	CMYK_M	CMYK_Y	CMYK_K	LAB_L	LAB_A	LAB_B	Opacity	Gloss	LAB_L_dry	LAB_A_dry
CMYK_C	1.000000e+00	4.791109e-02	4.791109e-02	4.201765e-02	-0.510016	-0.574932	-0.489379	8.165348e-17	1.689967e-16	-0.509297	-0.572220
CMYK_M	4.791109e-02	1.000000e+00	4.791109e-02	4.201765e-02	-0.544619	0.666733	-0.235930	-5.082038e-16	4.954927e-16	-0.547757	0.674593
CMYK_Y	4.791109e-02	4.791109e-02	1.000000e+00	4.201765e-02	-0.116207	-0.191877	0.712953	-2.548087e-17	-5.022469e-18	-0.110539	-0.192612
CMYK_K	4.201765e-02	4.201765e-02	4.201765e-02	1.000000e+00	-0.628491	-0.019264	-0.043703	1.432849e-16	-1.431810e-17	-0.629931	-0.021319
LAB_L	-5.100164e-01	-5.446191e-01	-1.162074e-01	-6.284908e-01	1.000000	-0.062277	0.381723	-6.432478e-03	1.831131e-03	0.999755	-0.067557
LAB_A	-5.749325e-01	6.667329e-01	-1.918772e-01	-1.926351e-02	-0.062277	1.000000	-0.022949	7.708927e-03	9.762502e-04	-0.064631	0.999616
LAB_B	-4.893791e-01	-2.359300e-01	7.129535e-01	-4.370284e-02	0.381723	-0.022949	1.000000	-6.467228e-04	2.294211e-03	0.386172	-0.026661
Opacity	8.165348e-17	-5.082038e-16	-2.548087e-17	1.432849e-16	-0.006432	0.007709	-0.000647	1.000000e+00	6.526002e-03	-0.007384	0.007570
Gloss	1.689967e-16	4.954927e-16	-5.022469e-18	-1.431810e-17	0.001831	0.000976	0.002294	6.526002e-03	1.000000e+00	0.006644	0.002046
LAB_L_dry	-5.092972e-01	-5.477566e-01	-1.105388e-01	-6.299307e-01	0.999755	-0.064631	0.386172	-7.383667e-03	6.644001e-03	1.000000	-0.069950
LAB_A_dry	-5.722203e-01	6.745928e-01	-1.926119e-01	-2.131924e-02	-0.067557	0.999616	-0.026661	7.569767e-03	2.045588e-03	-0.069950	1.000000
LAB_B_dry	-4.929648e-01	-2.366419e-01	7.183031e-01	-3.984945e-02	0.379304	-0.020963	0.999422	1.894877e-04	1.104404e-03	0.383871	-0.024868
Fluorescence	-1.325788e-17	-1.032318e-15	5.274283e-17	-9.435783e-16	-0.001032	-0.003465	-0.007356	-2.614840e-01	-4.069742e-01	-0.003274	-0.003872
Thickness	-1.705123e-17	5.620545e-17	5.107099e-17	-3.516921e-16	-0.008596	0.003356	0.001528	2.585280e-01	1.478490e-01	-0.006961	0.003964
Whiteness	2.963189e-17	1.306796e-17	2.964985e-17	-4.067976e-16	-0.011297	-0.006562	-0.007743	-3.742338e-01	-4.254368e-01	-0.013330	-0.007050
Smoothness	2.445862e-16	1.414000e-16	7.896916e-17	4.287705e-16	-0.005930	-0.004815	-0.006418	-3.311152e-01	-6.053176e-01	-0.009514	-0.005584
Bulk	1.389650e-15	6.554869e-17	-7.065454e-17	1.052554e-15	-0.004306	0.001944	-0.007242	2.469814e-01	-5.896744e-01	-0.007753	0.000952
Brightness	1.930363e-16	-2.226253e-16	-8.837147e-17	2.419819e-16	-0.000346	0.003489	0.007344	2.453881e-01	3.963542e-01	0.001700	0.003862
enc_Coating	-8.514066e-17	-1.990826e-16	-1.801217e-16	3.826006e-17	-0.001088	-0.003341	0.009338	7.870251e-02	-4.253592e-01	-0.001324	-0.003727
enc_Region	-1.826874e-16	-5.068304e-17	-9.139956e-16	-5.775103e-16	0.018526	0.005426	0.007486	3.837590e-01	4.488796e-01	0.020894	0.005976
enc_Supplier	-8.810753e-16	4.583209e-16	7.428271e-16	-4.555274e-16	0.012573	0.001467	0.006683	1.485469e-01	2.825654e-01	0.013051	0.001357

Figure D.1: Correlation Matrix (Part 1)

We observe that there is a very high correlation between the respective values of $L*$, $a*$, $b*$ at Time 00 and Time 96. There is a significant negative correlation between the $L*$, $a*$, $b*$ values at Time 96 and the C, M, Y and K (except for the 2 values). Generally speaking, the media characteristics (Section 5.3.2) do not seem any correlation with the C, M, Y and K values. This can be attributed to the fact that these values are not a property of media or ink but are values manually provided in the test chart. In fact, other than the $L*$, $a*$ and $b*$ values (at Time 00 and Time 96) the C, M, Y and K values do not have a correlation with any other quantity.

Generally speaking, the media characteristics have relatively higher correlation (in magnitude) among themselves compared to the correlation with other quantities such as $L*$, $a*$, $b*$, C, M, Y



	LAB_B_dry	Fluorescence	Thickness	Whiteness	Smoothness	Bulk	Brightness	enc_Coating	enc_Region	enc_Supplier
CMYK_C	-0.492965	-1.325788e-17	-1.705123e-17	2.963189e-17	2.445862e-16	1.389650e-15	1.930363e-16	-8.514066e-17	-1.826874e-16	-8.810753e-16
CMYK_M	-0.236642	-1.032318e-15	5.620545e-17	1.306796e-17	1.414000e-16	6.554869e-17	-2.226253e-16	-1.990826e-16	-5.068304e-17	4.583209e-16
CMYK_Y	0.718303	5.274283e-17	5.107099e-17	2.964985e-17	7.896916e-17	-7.065454e-17	-8.837147e-17	-1.801217e-16	-9.139956e-16	7.428271e-16
CMYK_K	-0.039849	-9.435783e-16	-3.516921e-16	-4.067976e-16	4.287705e-16	1.052554e-15	2.419819e-16	3.826006e-17	-5.775103e-16	-4.555274e-16
LAB_L	0.379304	-1.031655e-03	-8.595542e-03	-1.129702e-02	-5.930290e-03	-4.305732e-03	-3.463913e-04	-1.087769e-03	1.852603e-02	1.257298e-02
LAB_A	-0.020963	-3.465132e-03	3.356155e-03	-6.562183e-03	-4.815441e-03	1.943540e-03	3.489053e-03	-3.340634e-03	5.425933e-03	1.466940e-03
LAB_B	0.999422	-7.356467e-03	1.528224e-03	-7.742891e-03	-6.418455e-03	-7.242348e-03	7.344306e-03	9.338115e-03	7.485706e-03	6.683405e-03
Opacity	0.000189	-2.614840e-01	2.585280e-01	-3.742338e-01	-3.311152e-01	2.469814e-01	2.453881e-01	7.870251e-02	3.837590e-01	1.485469e-01
Gloss	0.001104	-4.069742e-01	1.478490e-01	-4.254368e-01	-6.053176e-01	-5.896744e-01	3.963542e-01	-4.253592e-01	4.488796e-01	2.825654e-01
LAB_L_dry	0.383871	-3.273881e-03	-6.961309e-03	-1.333000e-02	-9.514077e-03	-7.753129e-03	1.699689e-03	-1.323944e-03	2.089410e-02	1.305096e-02
LAB_A_dry	-0.024868	-3.871960e-03	3.963805e-03	-7.049914e-03	-5.583901e-03	9.521872e-04	3.861905e-03	-3.726654e-03	5.976167e-03	1.357174e-03
LAB_B_dry	1.000000	-6.844701e-03	1.414558e-03	-7.644089e-03	-5.722768e-03	-5.634765e-03	6.968130e-03	8.859327e-03	7.168265e-03	6.967486e-03
Fluorescence	-0.006845	1.000000e+00	-5.785928e-01	6.405219e-01	6.260559e-01	3.301460e-01	-9.966573e-01	-7.158707e-02	-5.680750e-01	-1.659011e-01
Thickness	0.001415	-5.785928e-01	1.000000e+00	-4.931277e-01	-4.528493e-01	-1.517244e-01	5.809713e-01	-1.590198e-03	2.803912e-01	-5.559133e-01
Whiteness	-0.007644	6.405219e-01	-4.931277e-01	1.000000e+00	8.960794e-01	4.610463e-01	-6.080449e-01	-5.664996e-02	-9.028808e-01	-1.416576e-01
Smoothness	-0.005723	6.260559e-01	-4.528493e-01	8.960794e-01	1.000000e+00	6.051536e-01	-5.940998e-01	-2.156455e-02	-8.100926e-01	-1.498341e-01
Bulk	-0.005635	3.301460e-01	-1.517244e-01	4.610463e-01	6.051536e-01	1.000000e+00	-3.072207e-01	-1.113589e-02	-4.183300e-01	-1.221694e-01
Brightness	0.006968	-9.966573e-01	5.809713e-01	-6.080449e-01	-5.940998e-01	-3.072207e-01	1.000000e+00	7.658567e-02	5.201364e-01	1.569907e-01
enc_Coating	0.008859	-7.158707e-02	-1.590198e-03	-5.664996e-02	-2.156455e-02	-1.113589e-02	7.658567e-02	1.000000e+00	-3.194383e-02	-4.353488e-02
enc_Region	0.007168	-5.680750e-01	2.803912e-01	-9.028808e-01	-8.100926e-01	-4.183300e-01	5.201364e-01	-3.194383e-02	1.000000e+00	3.991225e-01
enc_Supplier	0.006967	-1.659011e-01	-5.559133e-01	-1.416576e-01	-1.498341e-01	-1.221694e-01	1.569907e-01	-4.353488e-02	3.991225e-01	1.000000e+00

Figure D.2: Correlation Matrix (Part 2)

and K. This analysis helps us to understand how the different quantities are correlated to each other and also helps us understand the significance of each feature (or quantity). In other words, it conveys how much information about a quantity can be attained by measuring other quantity (or quantities).

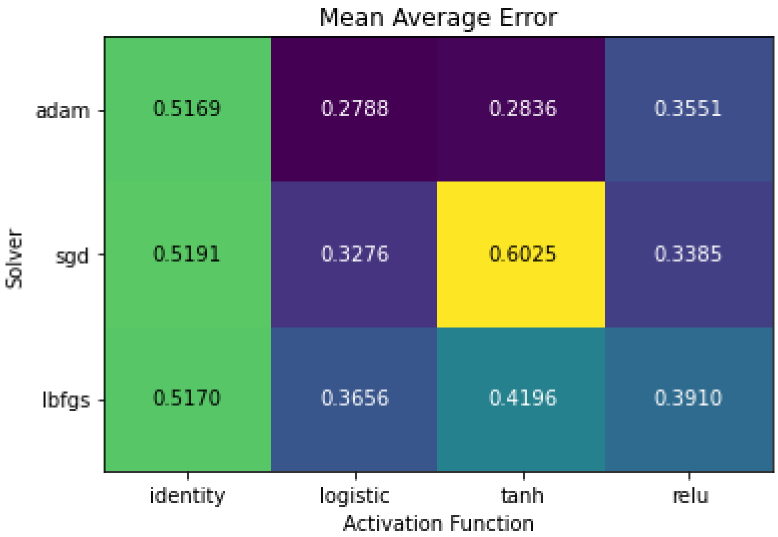
Appendix E

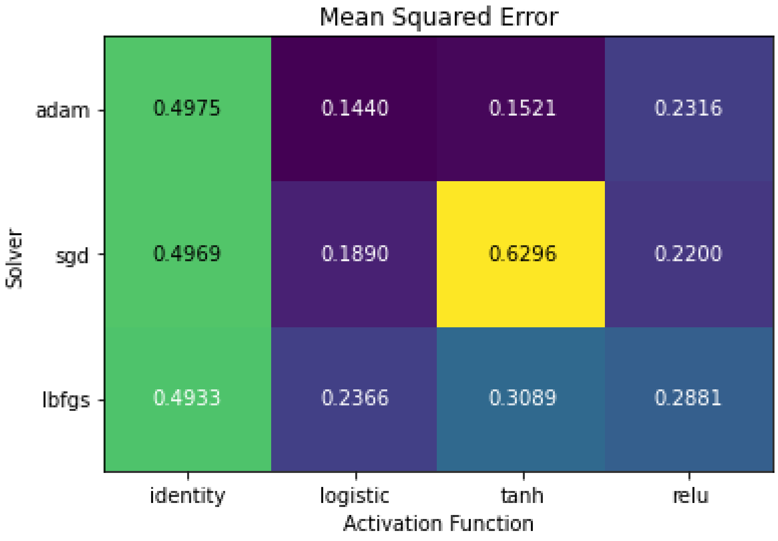
Model Performance Under Different Parameters

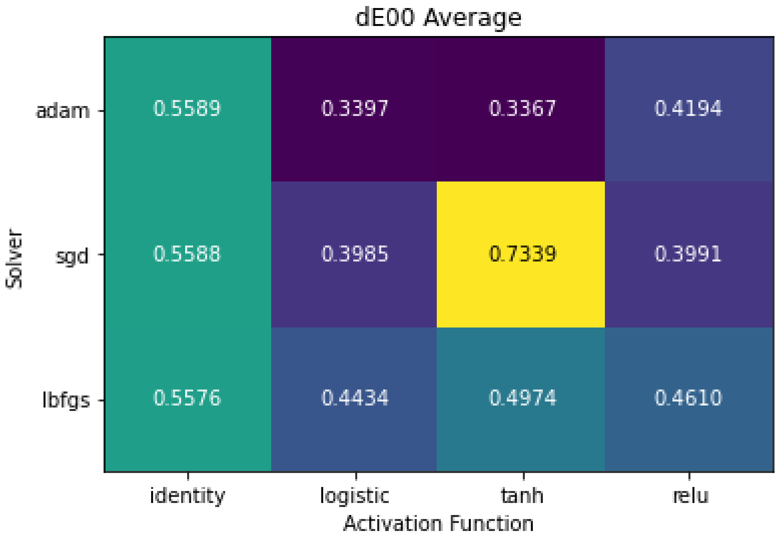
E.1 Multilayer Perceptron Model (MLP)

The following heatmaps are obtained on performing grid search on testing data with various solvers and activation functions of MLP regression algorithm.

Testing




Mean Squared Error
Solver
adam
sgd
lbfgs
Activation Function
identity
logistic
tanh
relu
0.4975
0.1440
0.1521
0.2316
0.4969
0.1890
0.6296
0.2200
0.4933
0.2366
0.3089
0.2881


dE00 Average
Solver
adam
sgd
lbfgs
Activation Function
identity
logistic
tanh
relu
0.5589
0.3397
0.3367
0.4194
0.5588
0.3985
0.7339
0.3991
0.5576
0.4434
0.4974
0.4610

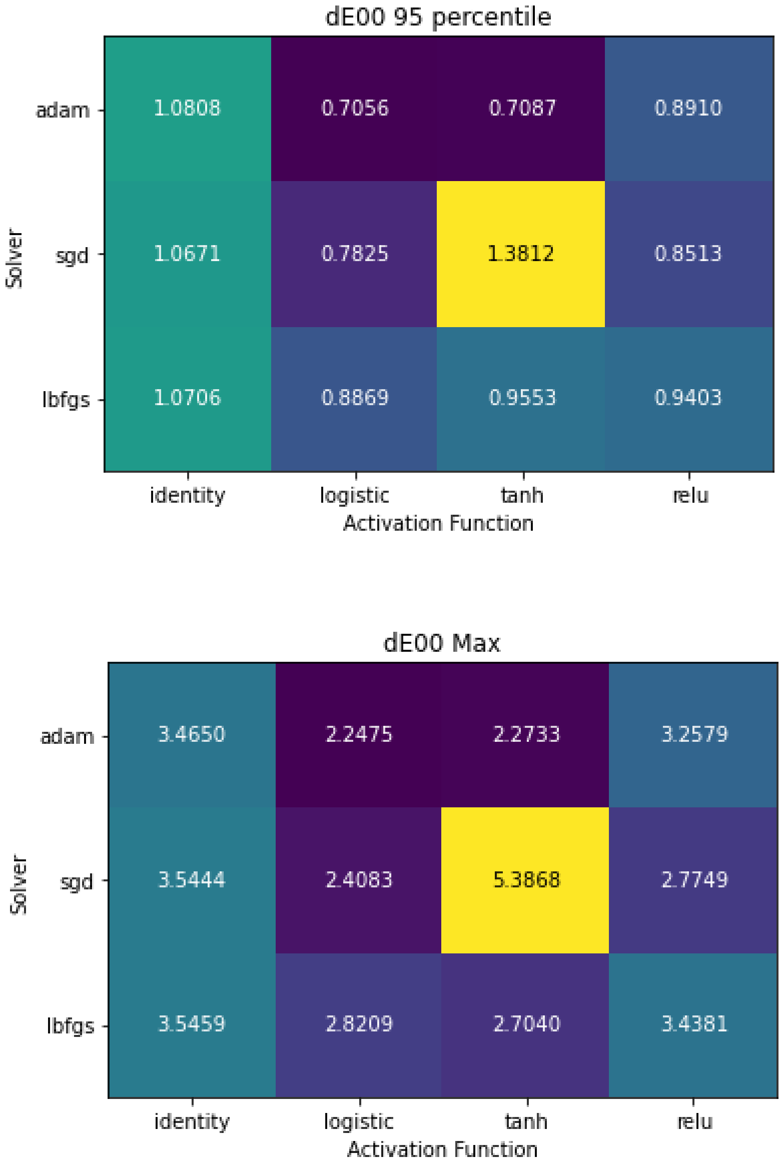


Figure E.1: Plot of Errors and δE on the Testing Data

E.2 Deep Neural Network

In this section, we provide the results obtained on the implementation of DNN model with various hyperparameters.

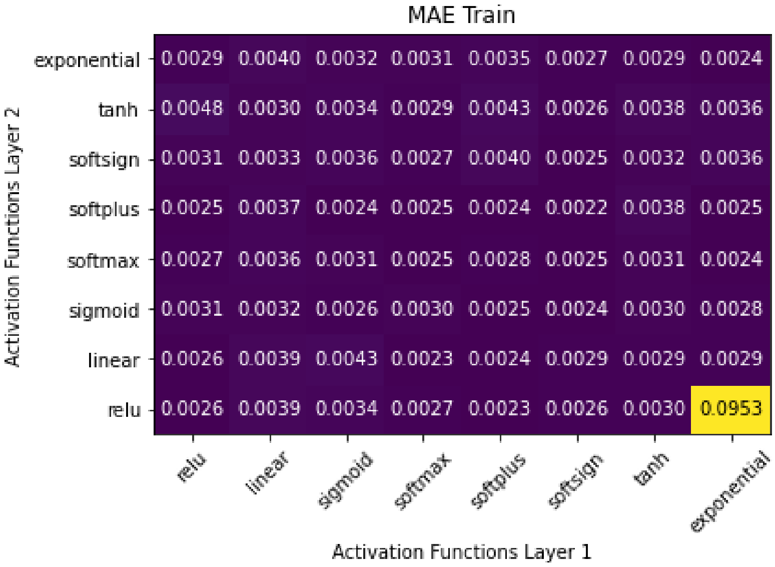
E.2.1 Activation Functions

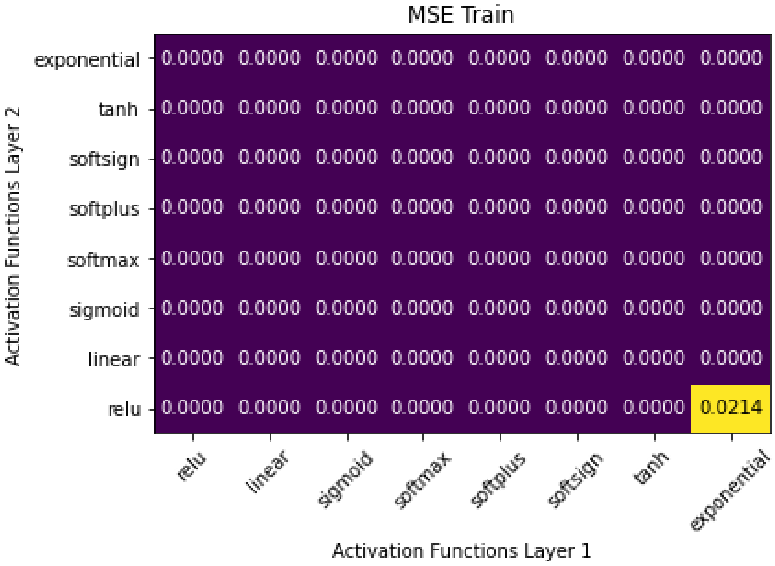
In this sub-section, we provide the results obtained on having linear and RELU activation functions in the final layer of the DNN model.

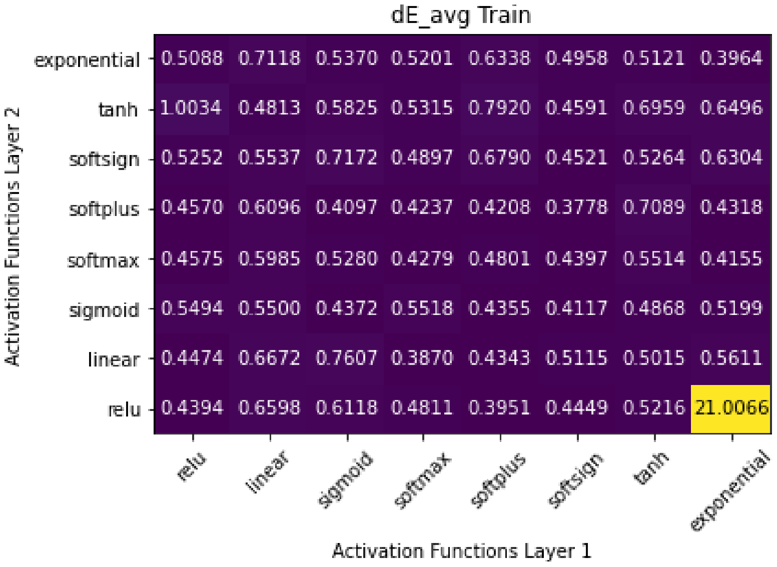
Linear Activation Function in Final Layer

Training

The following heatmaps are obtained on performing grid search on training data with various activation functions on first and second layer.

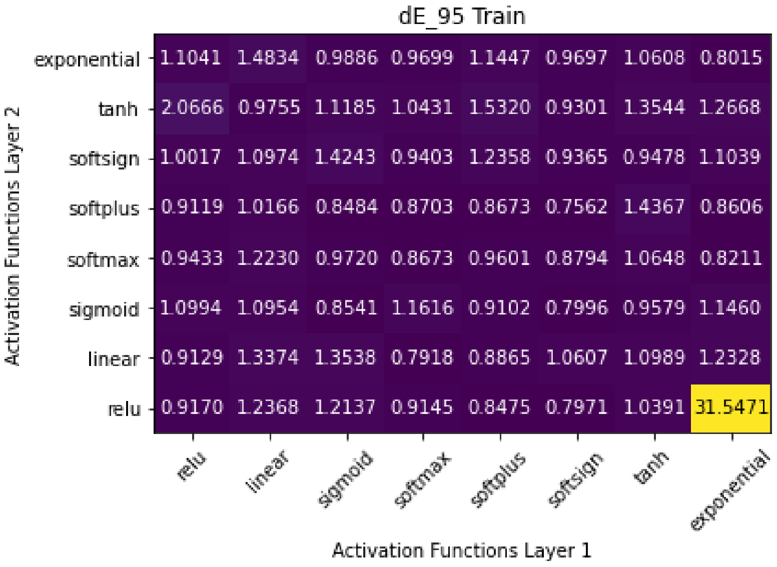






dE_avg Train

Activation Functions Layer 2 \ Activation Functions Layer 1	relu	linear	sigmoid	softmax	softplus	softsign	tanh	exponential
exponential	0.5088	0.7118	0.5370	0.5201	0.6338	0.4958	0.5121	0.3964
tanh	1.0034	0.4813	0.5825	0.5315	0.7920	0.4591	0.6959	0.6496
softsign	0.5252	0.5537	0.7172	0.4897	0.6790	0.4521	0.5264	0.6304
softplus	0.4570	0.6096	0.4097	0.4237	0.4208	0.3778	0.7089	0.4318
softmax	0.4575	0.5985	0.5280	0.4279	0.4801	0.4397	0.5514	0.4155
sigmoid	0.5494	0.5500	0.4372	0.5518	0.4355	0.4117	0.4868	0.5199
linear	0.4474	0.6672	0.7607	0.3870	0.4343	0.5115	0.5015	0.5611
relu	0.4394	0.6598	0.6118	0.4811	0.3951	0.4449	0.5216	21.0066



dE_95 Train

Activation Functions Layer 2 \ Activation Functions Layer 1	relu	linear	sigmoid	softmax	softplus	softsign	tanh	exponential
exponential	1.1041	1.4834	0.9886	0.9699	1.1447	0.9697	1.0608	0.8015
tanh	2.0666	0.9755	1.1185	1.0431	1.5320	0.9301	1.3544	1.2668
softsign	1.0017	1.0974	1.4243	0.9403	1.2358	0.9365	0.9478	1.1039
softplus	0.9119	1.0166	0.8484	0.8703	0.8673	0.7562	1.4367	0.8606
softmax	0.9433	1.2230	0.9720	0.8673	0.9601	0.8794	1.0648	0.8211
sigmoid	1.0994	1.0954	0.8541	1.1616	0.9102	0.7996	0.9579	1.1460
linear	0.9129	1.3374	1.3538	0.7918	0.8865	1.0607	1.0989	1.2328
relu	0.9170	1.2368	1.2137	0.9145	0.8475	0.7971	1.0391	31.5471

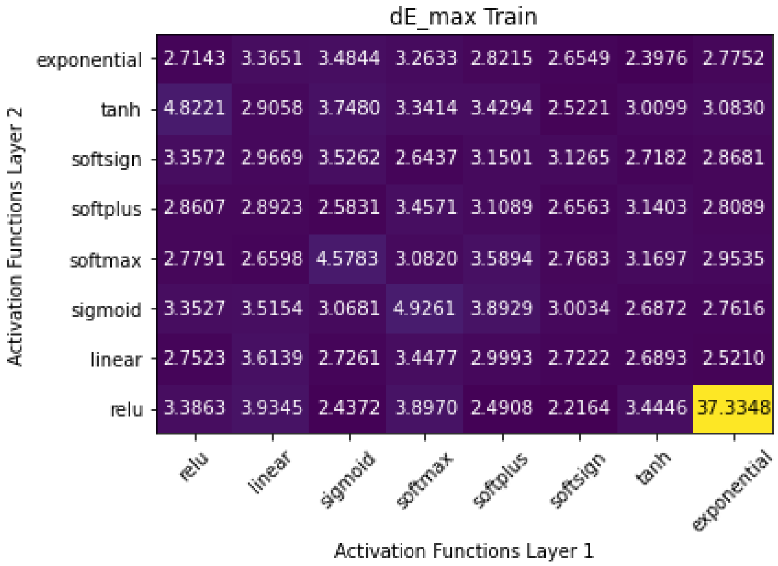
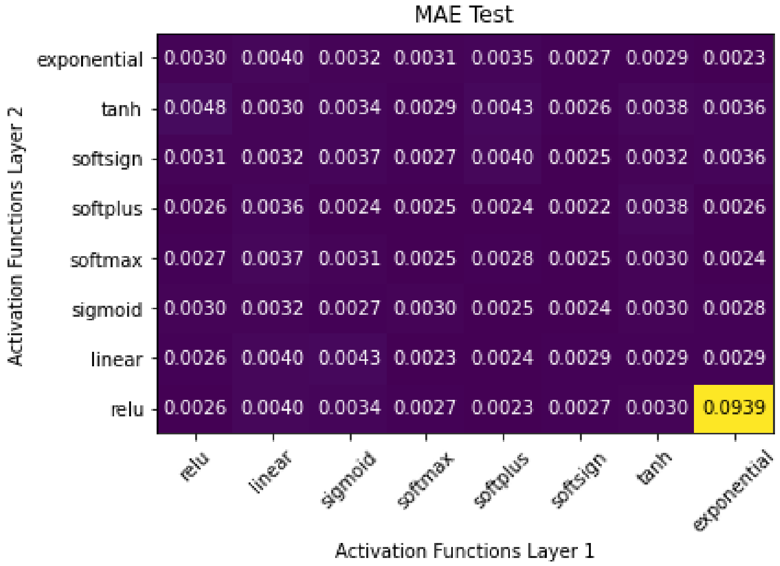
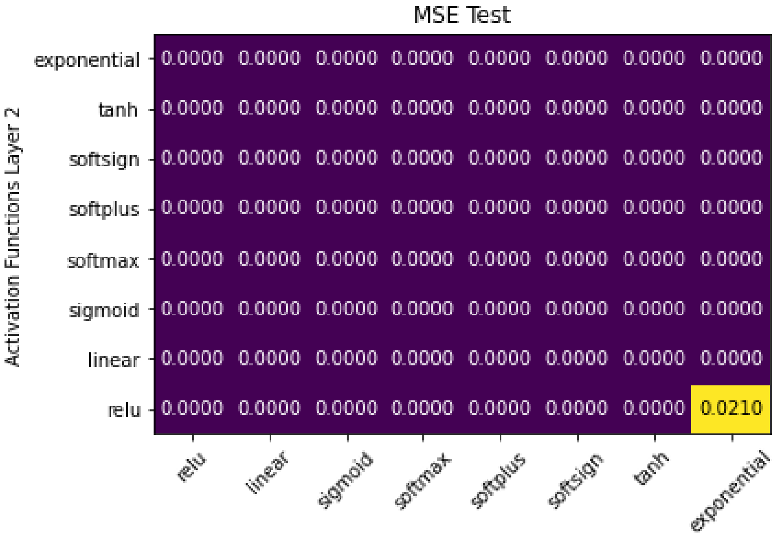


Figure E.2: Plot of Errors and δE on the Training Data on Linear Activation Function

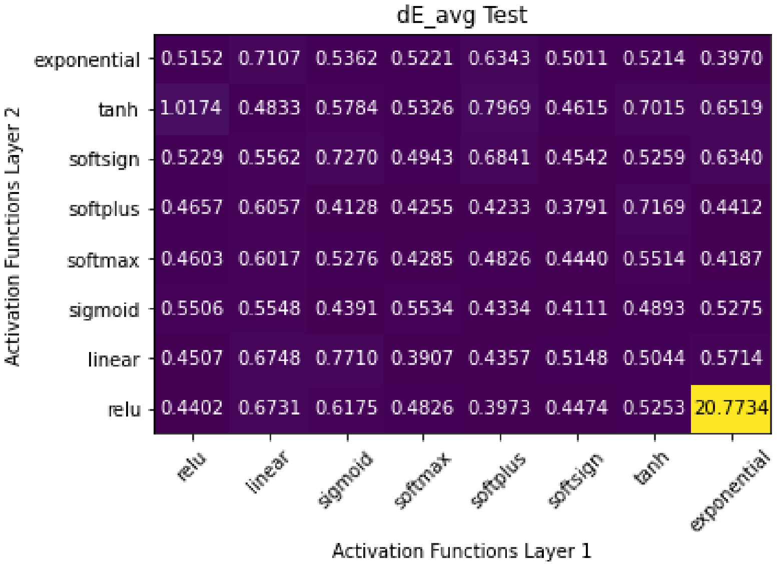
Testing

The following heatmaps are obtained on performing grid search on testing data with various activation functions on first and second layer.





	relu	linear	sigmoid	softmax	softplus	softsign	tanh	exponential
exponential	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
tanh	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
softsign	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
softplus	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
softmax	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
sigmoid	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
linear	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
relu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0210



	relu	linear	sigmoid	softmax	softplus	softsign	tanh	exponential
exponential	0.5152	0.7107	0.5362	0.5221	0.6343	0.5011	0.5214	0.3970
tanh	1.0174	0.4833	0.5784	0.5326	0.7969	0.4615	0.7015	0.6519
softsign	0.5229	0.5562	0.7270	0.4943	0.6841	0.4542	0.5259	0.6340
softplus	0.4657	0.6057	0.4128	0.4255	0.4233	0.3791	0.7169	0.4412
softmax	0.4603	0.6017	0.5276	0.4285	0.4826	0.4440	0.5514	0.4187
sigmoid	0.5506	0.5548	0.4391	0.5534	0.4334	0.4111	0.4893	0.5275
linear	0.4507	0.6748	0.7710	0.3907	0.4357	0.5148	0.5044	0.5714
relu	0.4402	0.6731	0.6175	0.4826	0.3973	0.4474	0.5253	20.7734

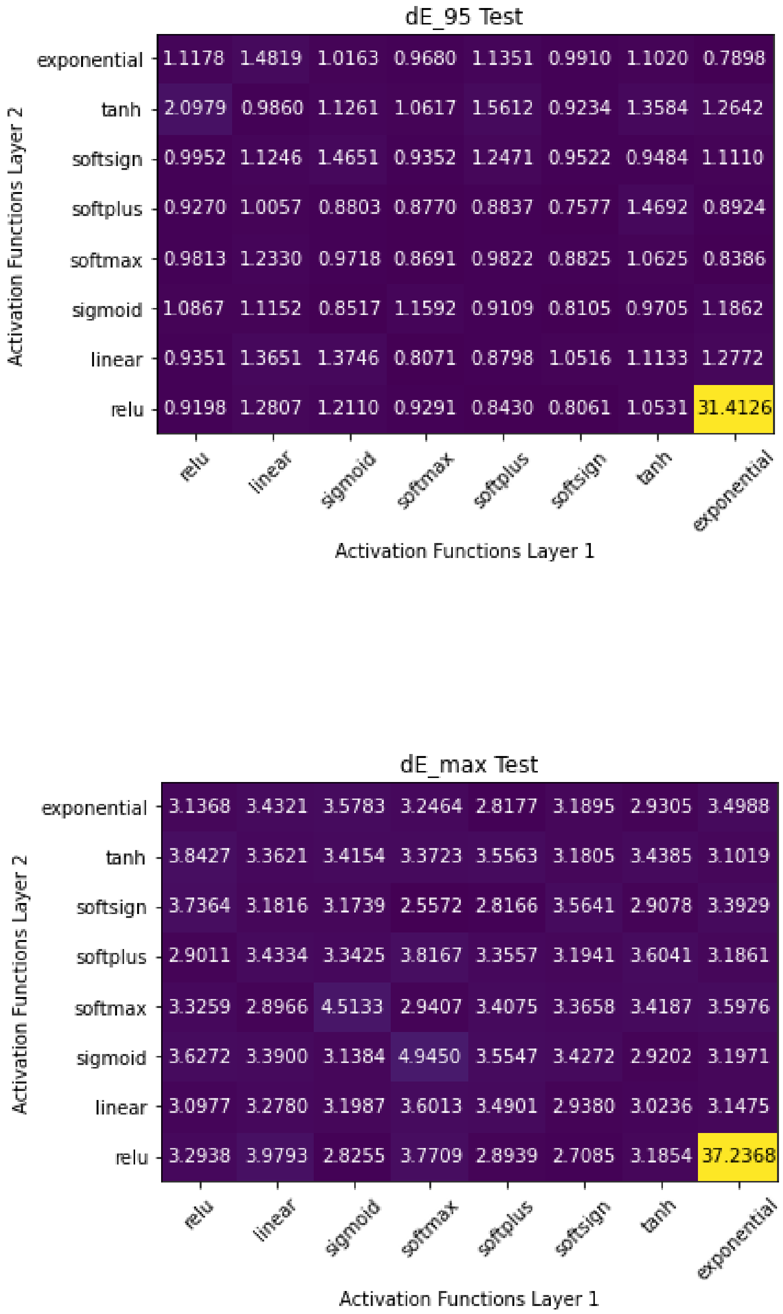
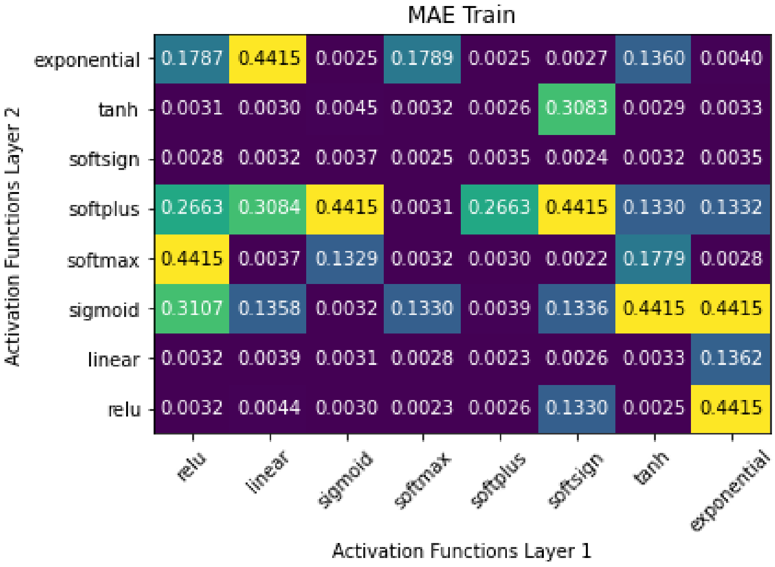
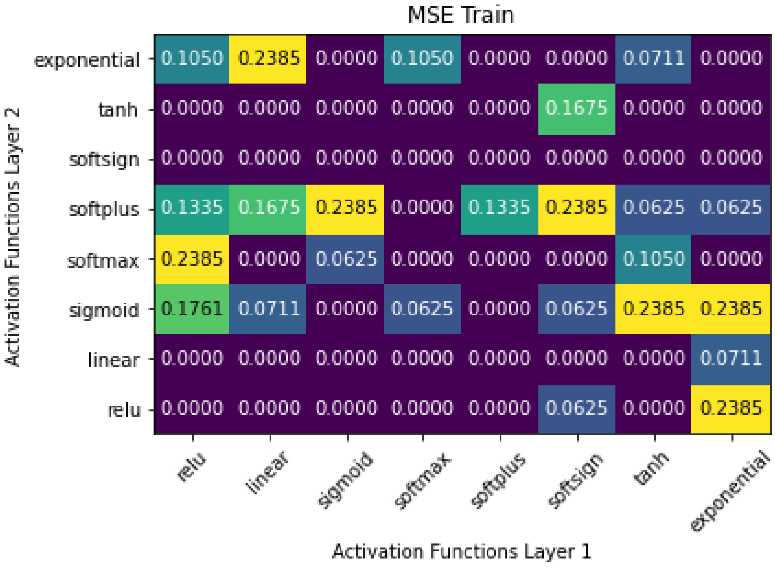


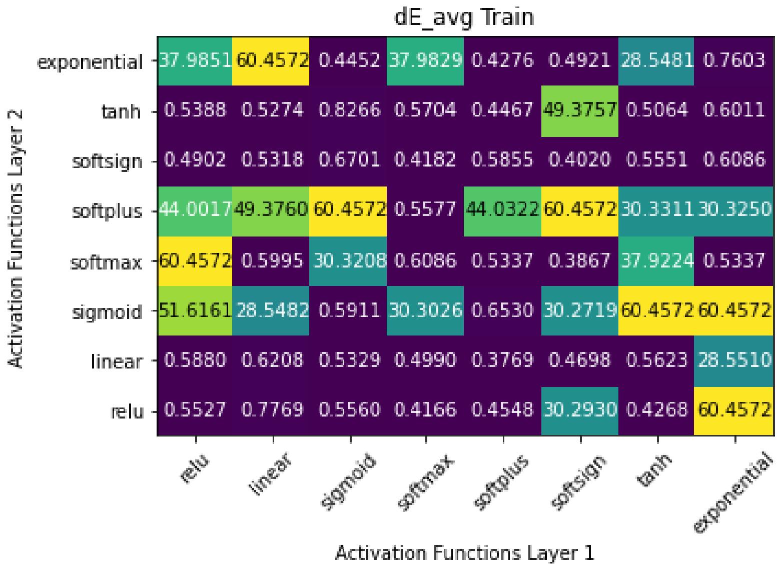
Figure E.3: Plot of Errors and δE on the Testing Data on Linear Activation Function

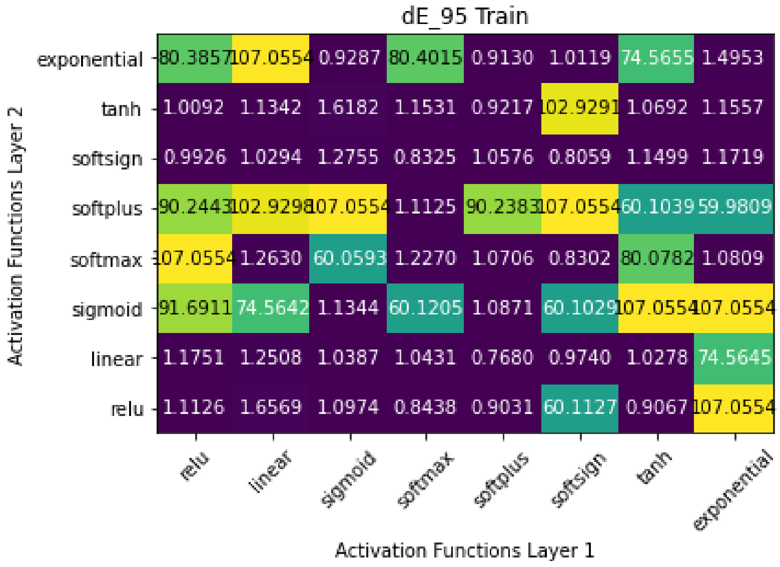
Relu Activation Function in Final Layer

Training The following heatmaps are obtained on performing grid search on training data with various activation functions on first and second layer.


MAE Train
Activation Functions Layer 2
exponential 0.1787 0.4415 0.0025 0.1789 0.0025 0.0027 0.1360 0.0040
tanh 0.0031 0.0030 0.0045 0.0032 0.0026 0.3083 0.0029 0.0033
softsign 0.0028 0.0032 0.0037 0.0025 0.0035 0.0024 0.0032 0.0035
softplus 0.2663 0.3084 0.4415 0.0031 0.2663 0.4415 0.1330 0.1332
softmax 0.4415 0.0037 0.1329 0.0032 0.0030 0.0022 0.1779 0.0028
sigmoid 0.3107 0.1358 0.0032 0.1330 0.0039 0.1336 0.4415 0.4415
linear 0.0032 0.0039 0.0031 0.0028 0.0023 0.0026 0.0033 0.1362
relu 0.0032 0.0044 0.0030 0.0023 0.0026 0.1330 0.0025 0.4415
relu linear sigmoid softmax softplus softsign tanh exponential
Activation Functions Layer 1


MSE Train
Activation Functions Layer 2
exponential 0.1050 0.2385 0.0000 0.1050 0.0000 0.0000 0.0711 0.0000
tanh 0.0000 0.0000 0.0000 0.0000 0.0000 0.1675 0.0000 0.0000
softsign 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
softplus 0.1335 0.1675 0.2385 0.0000 0.1335 0.2385 0.0625 0.0625
softmax 0.2385 0.0000 0.0625 0.0000 0.0000 0.0000 0.1050 0.0000
sigmoid 0.1761 0.0711 0.0000 0.0625 0.0000 0.0625 0.2385 0.2385
linear 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0711
relu 0.0000 0.0000 0.0000 0.0000 0.0000 0.0625 0.0000 0.2385
relu linear sigmoid softmax softplus softsign tanh exponential
Activation Functions Layer 1


dE_avg Train
Activation Functions Layer 2
exponential
tanh
softsign
softplus
softmax
sigmoid
linear
relu
37.9851 60.4572 0.4452 37.9829 0.4276 0.4921 28.5481 0.7603
0.5388 0.5274 0.8266 0.5704 0.4467 49.3757 0.5064 0.6011
0.4902 0.5318 0.6701 0.4182 0.5855 0.4020 0.5551 0.6086
44.0017 49.3760 60.4572 0.5577 44.0322 60.4572 30.3311 30.3250
60.4572 0.5995 30.3208 0.6086 0.5337 0.3867 37.9224 0.5337
51.6161 28.5482 0.5911 30.3026 0.6530 30.2719 60.4572 60.4572
0.5880 0.6208 0.5329 0.4990 0.3769 0.4698 0.5623 28.5510
0.5527 0.7769 0.5560 0.4166 0.4548 30.2930 0.4268 60.4572
relu
linear
sigmoid
softmax
softplus
softsign
tanh
exponential
Activation Functions Layer 1


dE_95 Train
Activation Functions Layer 2
exponential
tanh
softsign
softplus
softmax
sigmoid
linear
relu
80.3857 107.0554 0.9287 80.4015 0.9130 1.0119 74.5655 1.4953
1.0092 1.1342 1.6182 1.1531 0.9217 102.9291 1.0692 1.1557
0.9926 1.0294 1.2755 0.8325 1.0576 0.8059 1.1499 1.1719
90.2443 102.9298 107.0554 1.1125 90.2383 107.0554 60.1039 59.9809
107.0554 1.2630 60.0593 1.2270 1.0706 0.8302 80.0782 1.0809
91.6911 74.5642 1.1344 60.1205 1.0871 60.1029 107.0554 107.0554
1.1751 1.2508 1.0387 1.0431 0.7680 0.9740 1.0278 74.5645
1.1126 1.6569 1.0974 0.8438 0.9031 60.1127 0.9067 107.0554
relu
linear
sigmoid
softmax
softplus
softsign
tanh
exponential
Activation Functions Layer 1

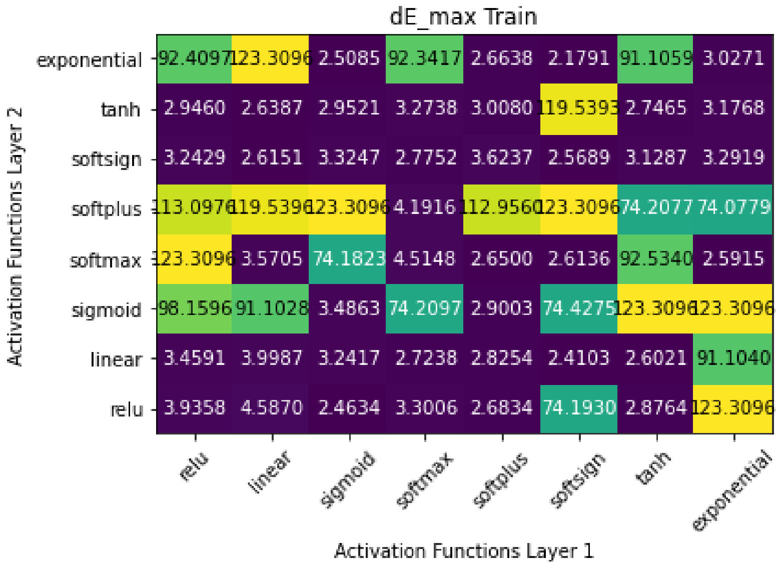
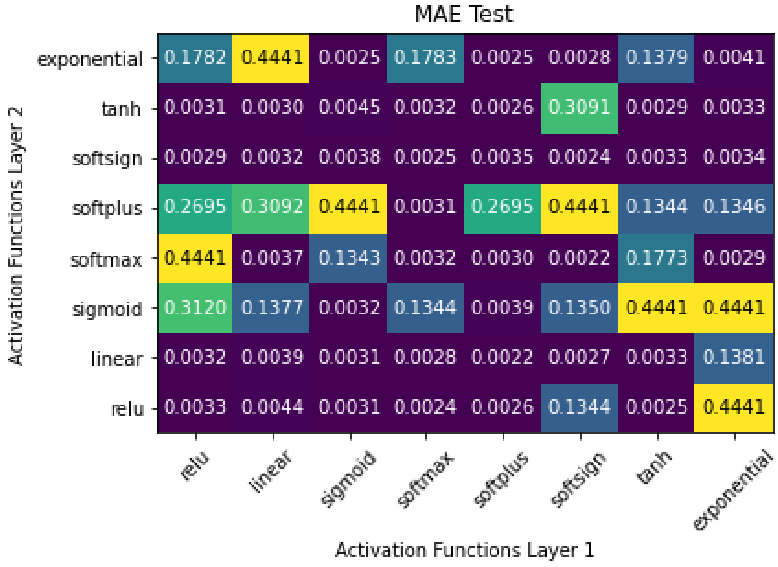
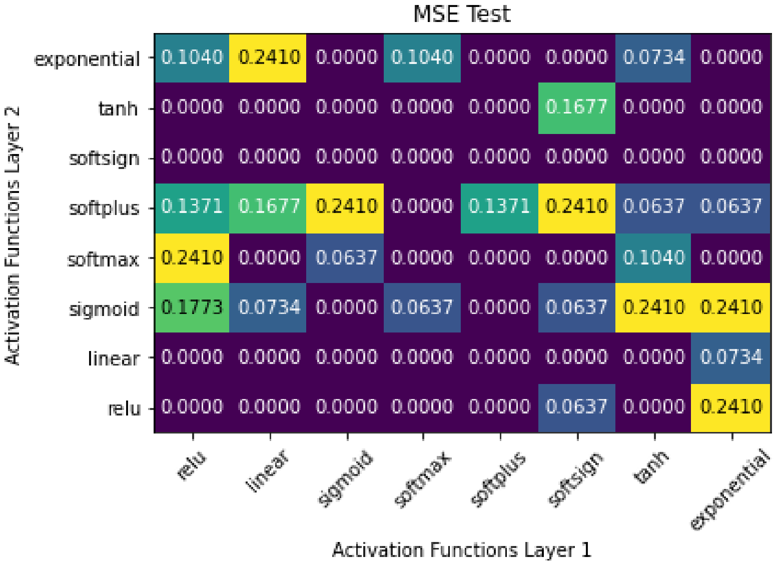


Figure E.4: Plot of Errors and δE on the Training Data on Relu Activation Function

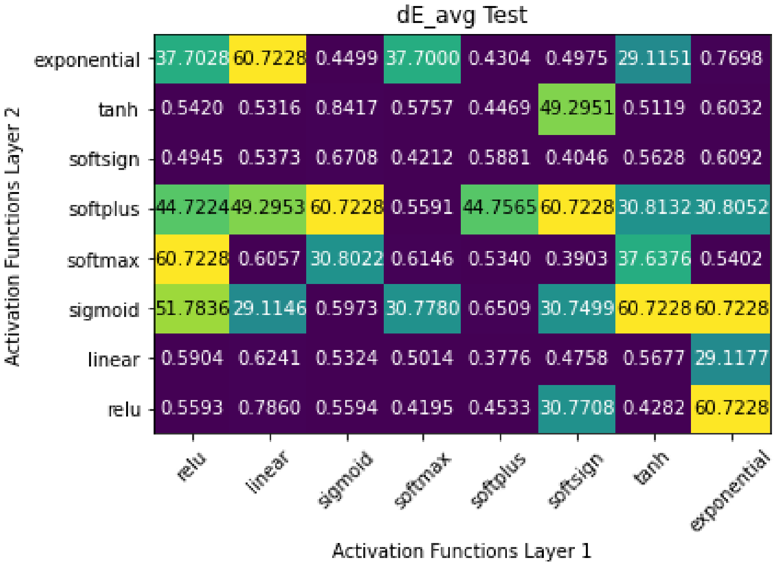
Testing

The following heatmaps are obtained on performing grid search on testing data with various activation functions on first and second layer.





MSE Test



dE_avg Test

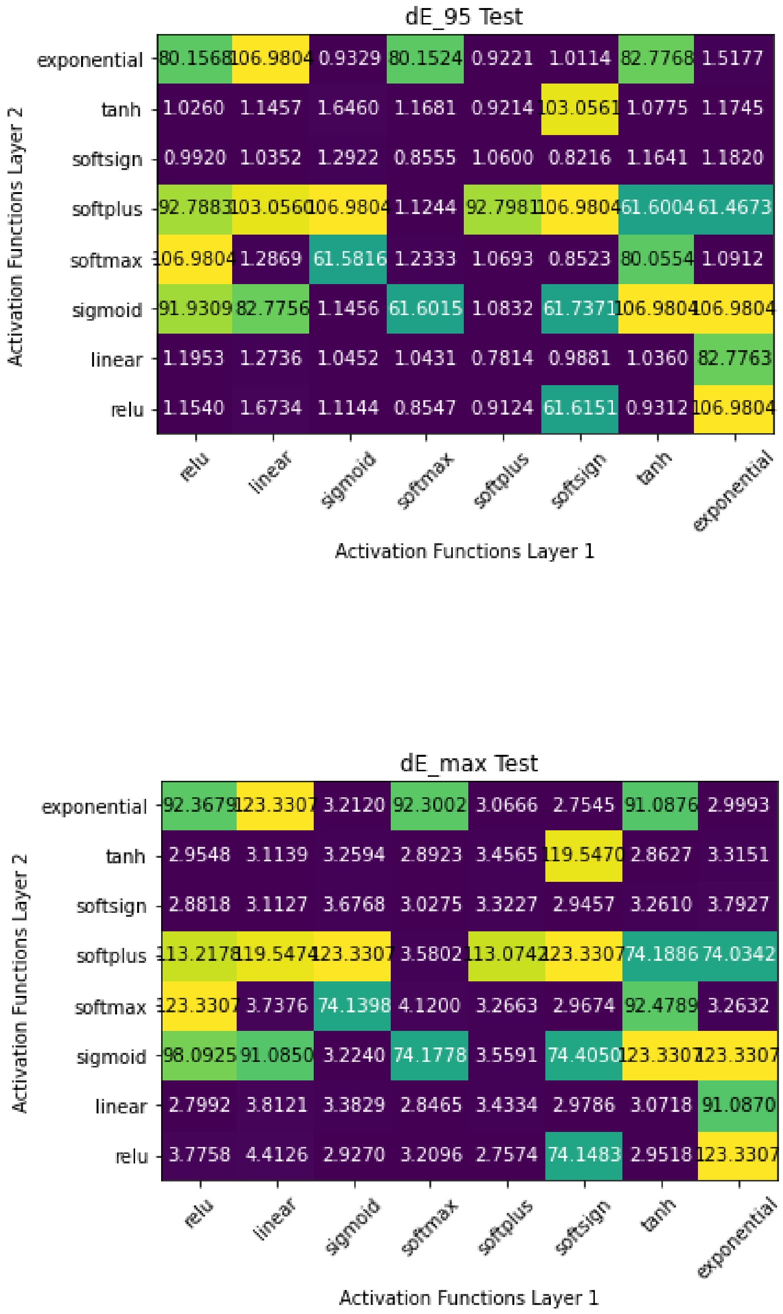
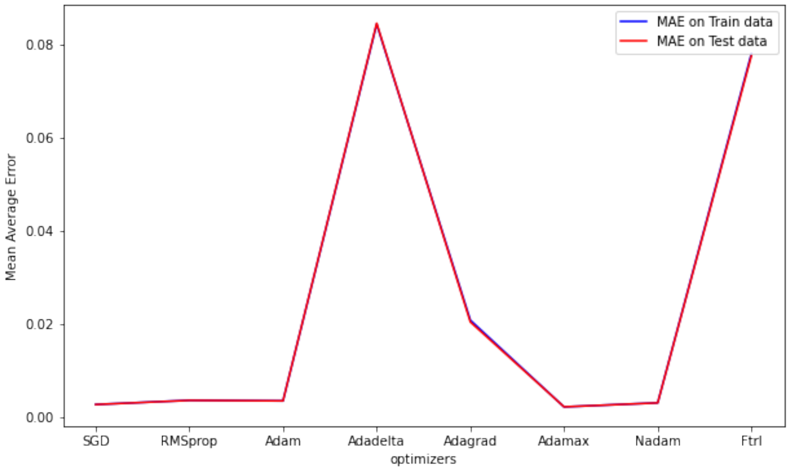
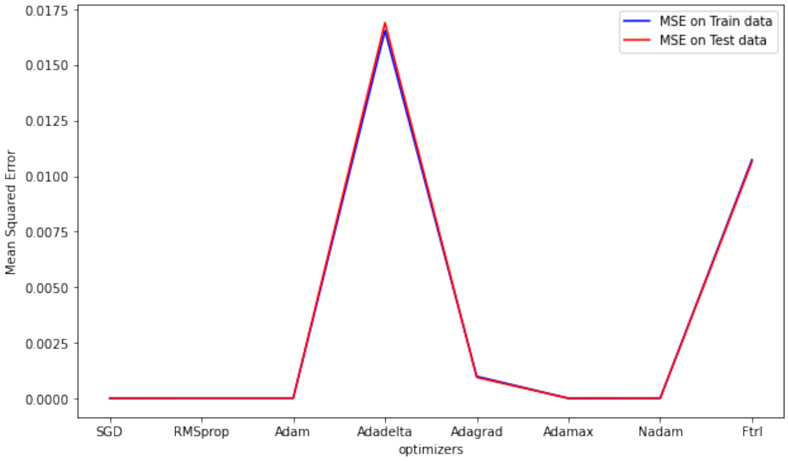


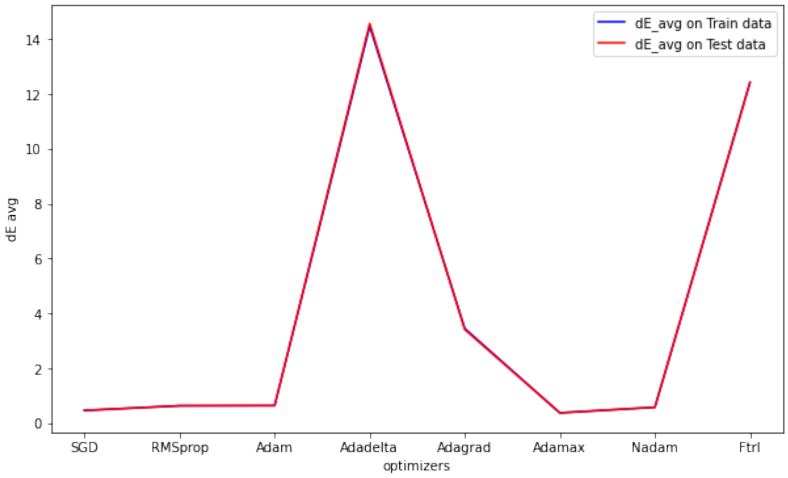
Figure E.5: Plot of Errors and δE on the Testing Data on Relu Activation Function

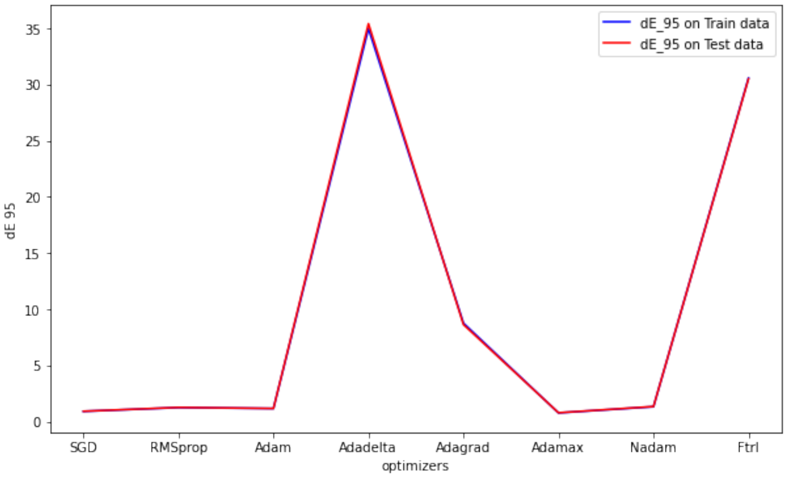
E.2.2 Optimizers

In this sub-section, we provide the results obtained on having different optimizers in the DNN model.


MAE on Train data
MAE on Test data
Mean Average Error
0.00
0.02
0.04
0.06
0.08
SGD
RMSprop
Adam
Adadelta
Adagrad
Adamax
Nadam
Ftrl
optimizers


MSE on Train data
MSE on Test data
Mean Squared Error
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
SGD
RMSprop
Adam
Adadelta
Adagrad
Adamax
Nadam
Ftrl
optimizers


dE_avg on Train data
dE_avg on Test data
dE avg
0
2
4
6
8
10
12
14
SGD
RMSprop
Adam
Adadelta
Adagrad
Adamax
Nadam
Ftrl
optimizers



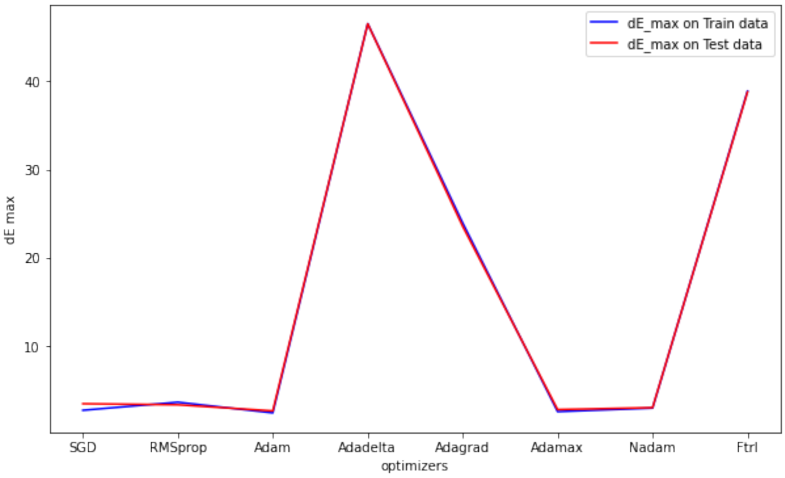
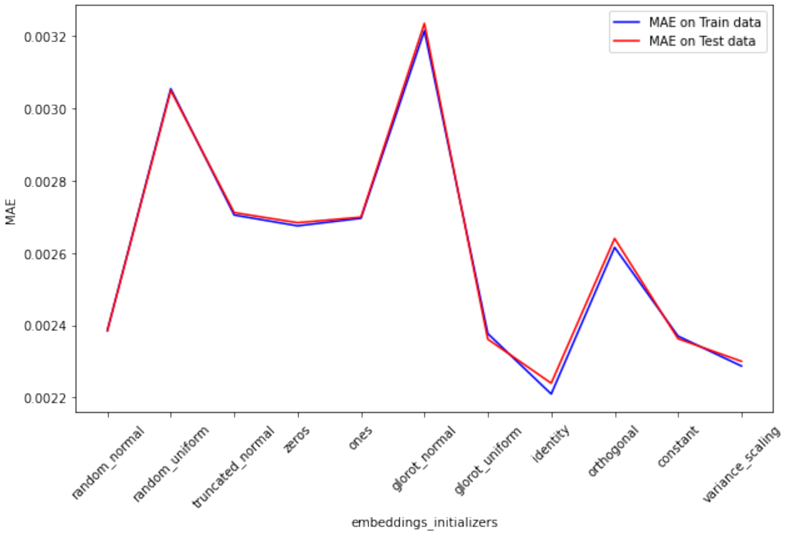
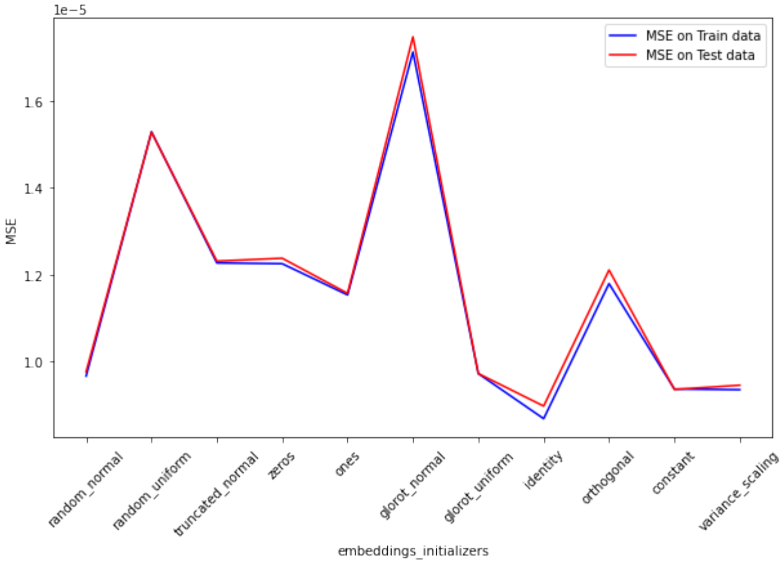


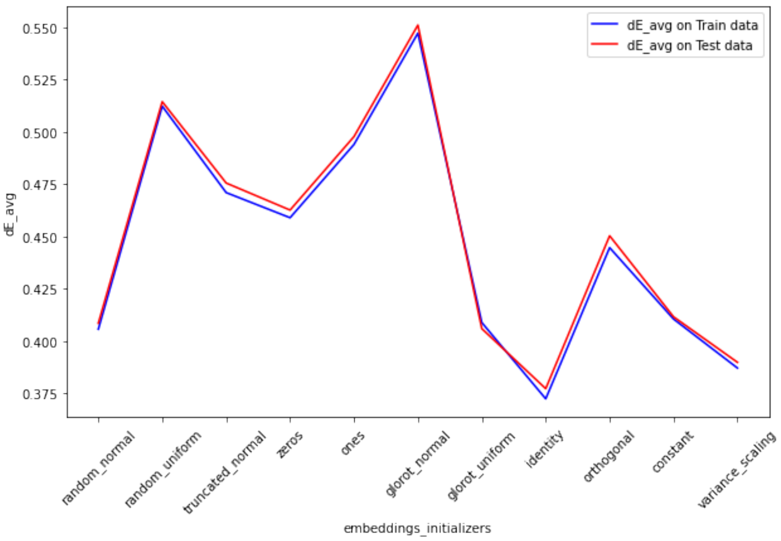
Figure E.6: Plot of Errors and δE on the Training and Testing Data with Over Different Optimizers

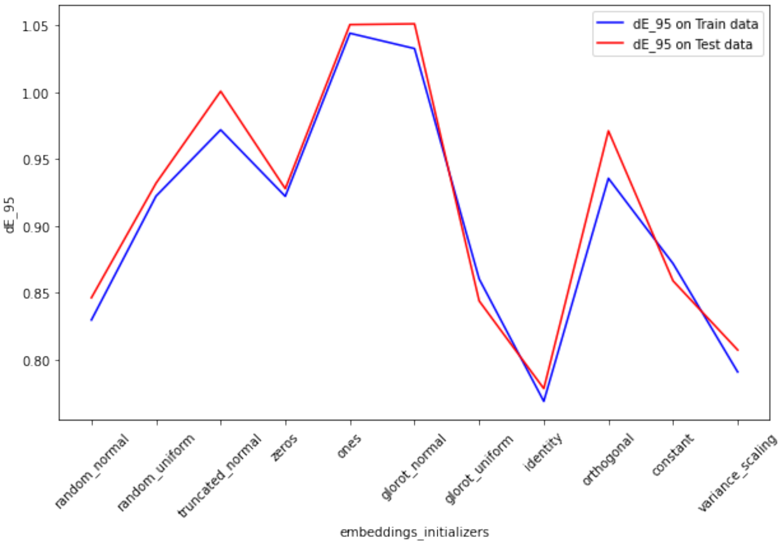
E.2.3 Embedding Initilizers

In this sub-section, we provide the results obtained on having different embedding initializers on all the embedding layers in the DNN model.


MAE on Train data
MAE on Test data
0.0032
0.0030
0.0028
0.0026
0.0024
0.0022
MAE
random_normal
random_uniform
truncated_normal
zeros
ones
glorot_normal
glorot_uniform
identity
orthogonal
constant
variance_scaling
embeddings_initializers


1e−5
MSE on Train data
MSE on Test data
1.6
1.4
1.2
1.0
MSE
random_normal
random_uniform
truncated_normal
zeros
ones
glorot_normal
glorot_uniform
identity
orthogonal
constant
variance_scaling
embeddings_initializers


dE_avg on Train data
dE_avg on Test data
dE_avg
0.550
0.525
0.500
0.475
0.450
0.425
0.400
0.375
random_normal
random_uniform
truncated_normal
zeros
ones
glorot_normal
glorot_uniform
identity
orthogonal
constant
variance_scaling
embeddings_initializers


dE_95 on Train data
dE_95 on Test data
dE_95
1.05
1.00
0.95
0.90
0.85
0.80
random_normal
random_uniform
truncated_normal
zeros
ones
glorot_normal
glorot_uniform
identity
orthogonal
constant
variance_scaling
embeddings_initializers

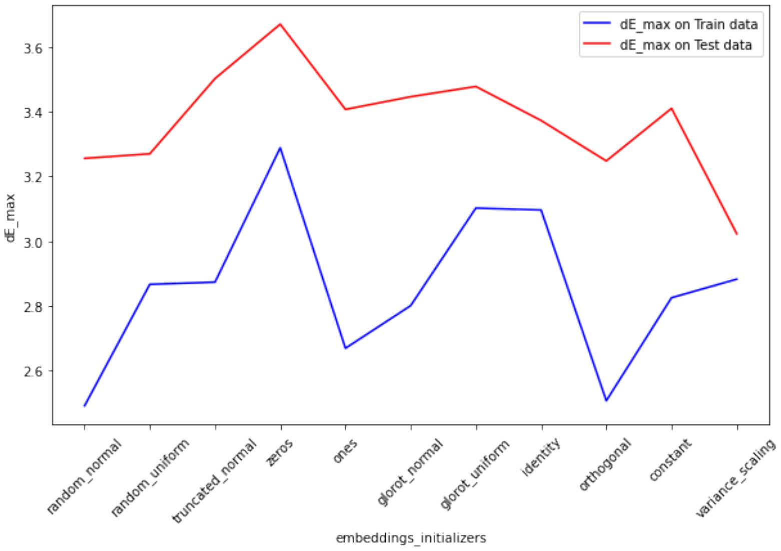
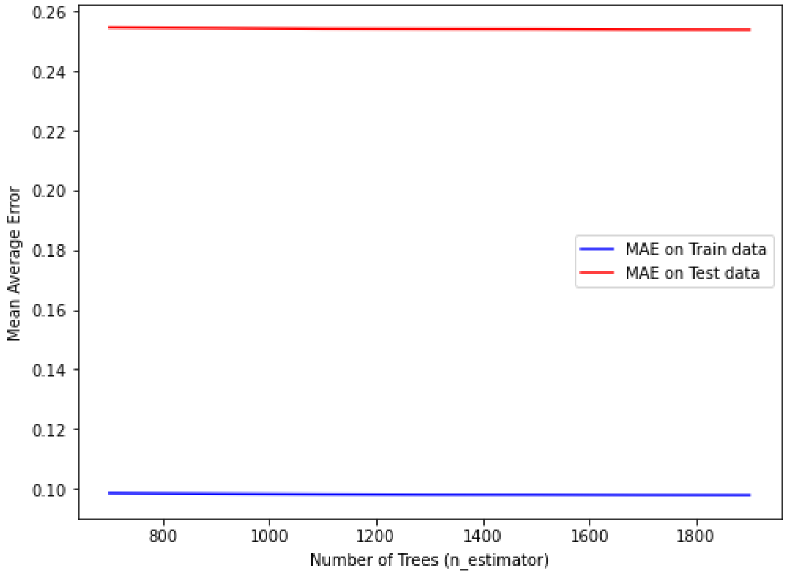
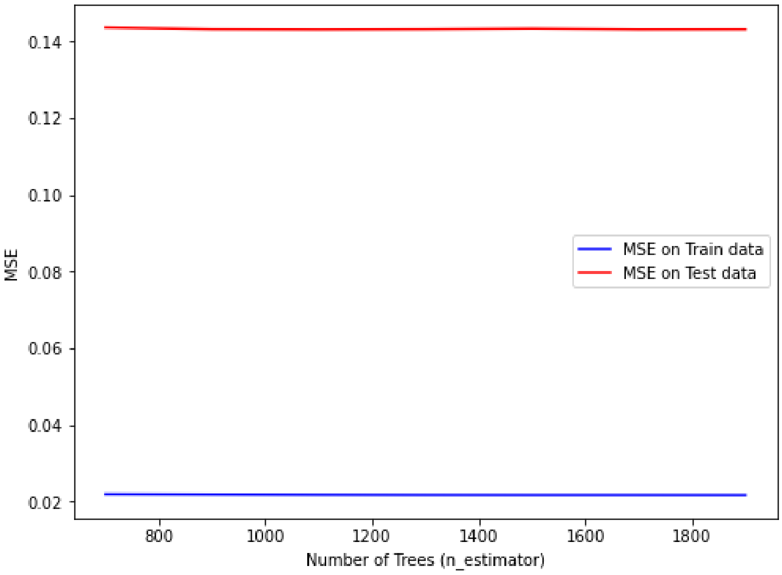


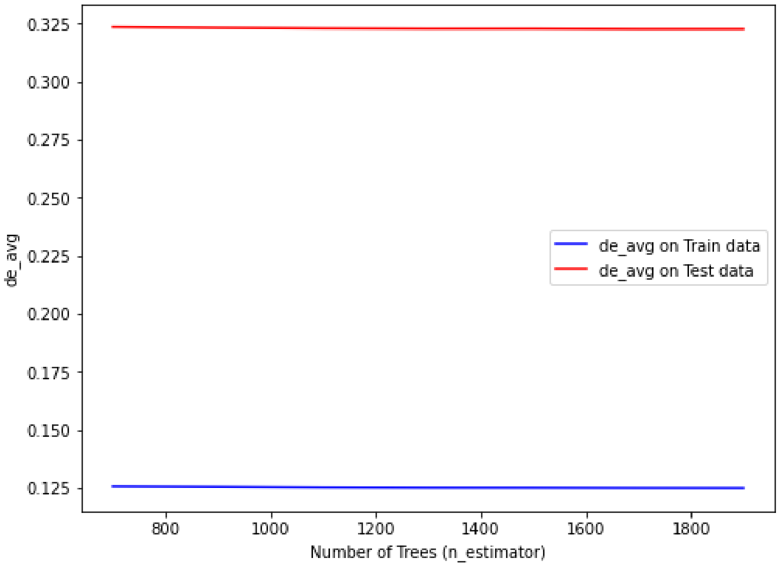
Figure E.7: Plot of Errors and δE on the Training and Testing Data

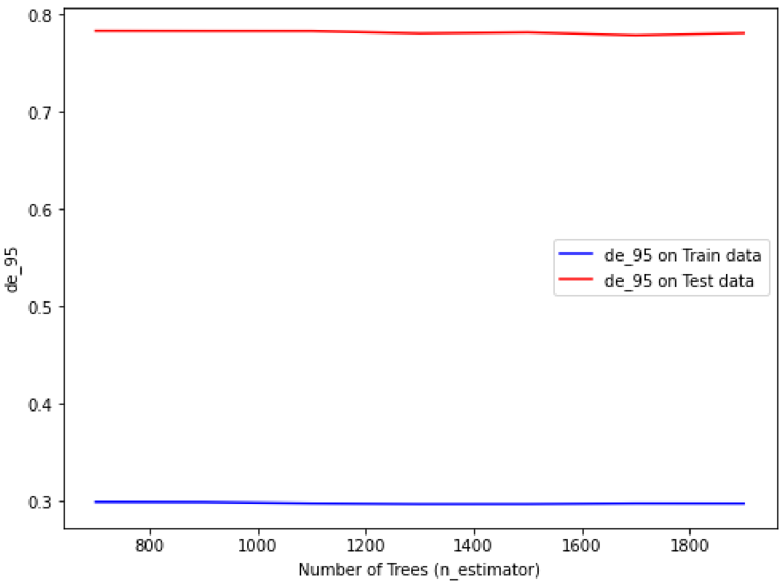
E.3 Random Forest (RF) Regression

In this section, we provide the results obtained on the having different *n_estimator* for Random Forest algorithm.




MSE
0.14
0.12
0.10
0.08
0.06
0.04
0.02
800
1000
1200
1400
1600
1800
Number of Trees (n_estimator)
MSE on Train data
MSE on Test data


de_avg
0.325
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
800
1000
1200
1400
1600
1800
Number of Trees (n_estimator)
de_avg on Train data
de_avg on Test data



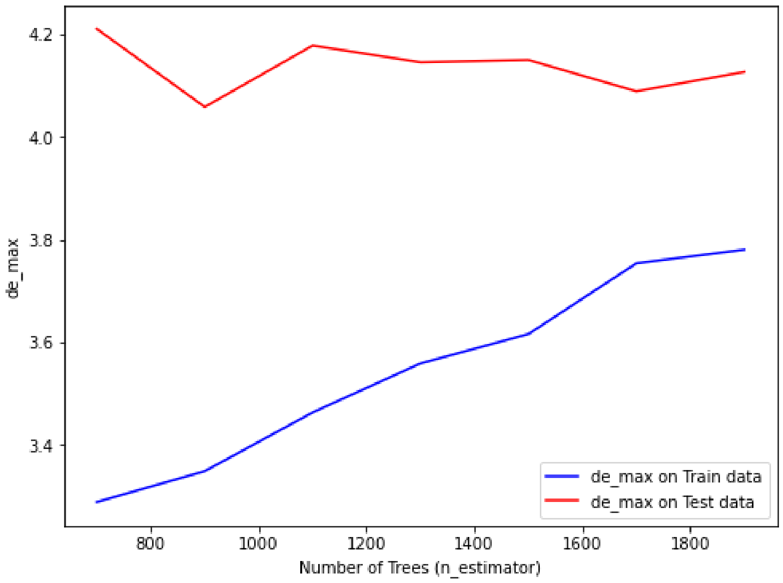
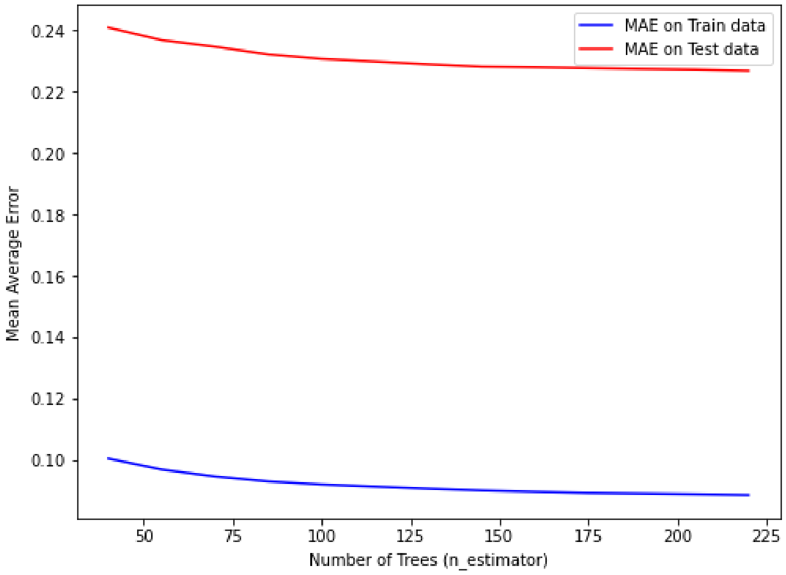


Figure E.8: Plot of Errors and δE on the Training and Testing Data for Random Forest

E.4 Extra Tree (ET) Regression

In this section, we provide the results obtained on the having different *n_estimator* for Extra Trees algorithm.



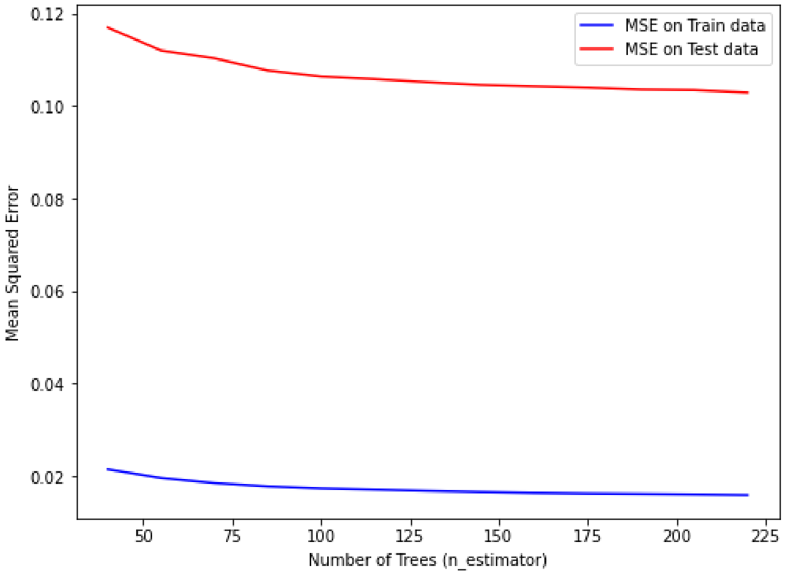


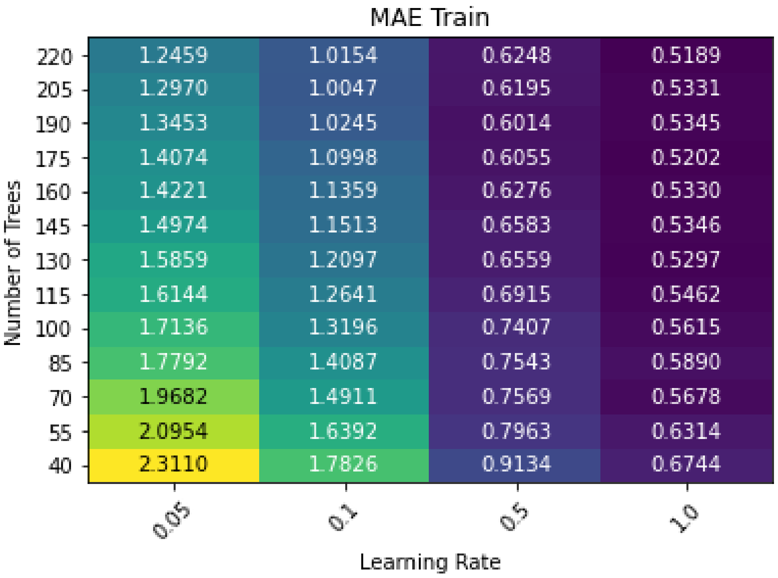
Figure E.9: Plot of Errors and δE on the Training and Testing Data

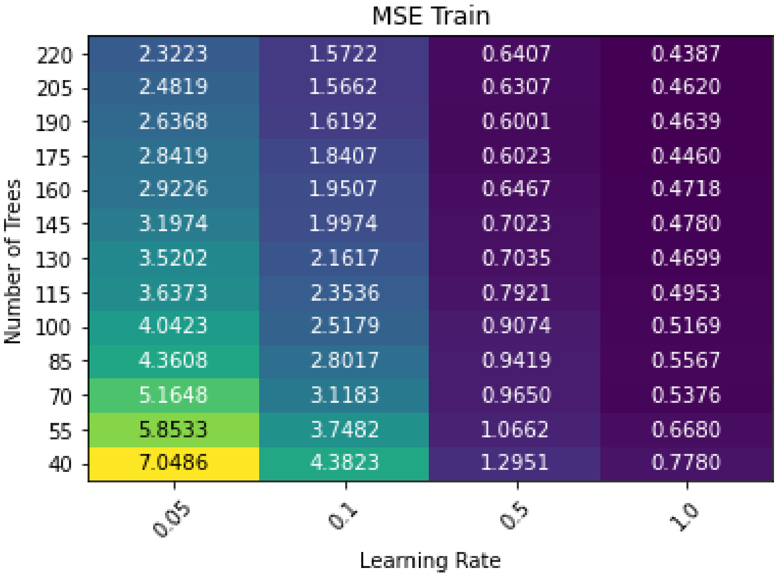
E.5 Adaptive Boost (AB) Regression

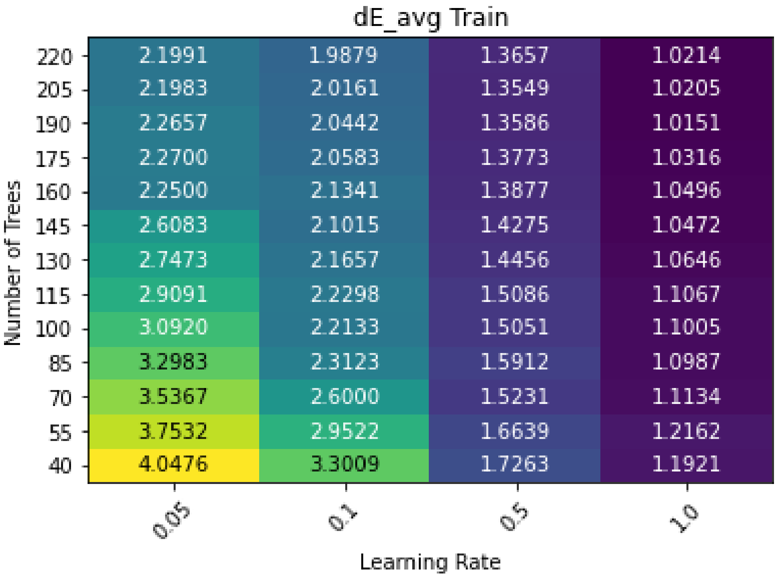
In this section, we provide the results obtained on performing grid search with different hyperparameters.

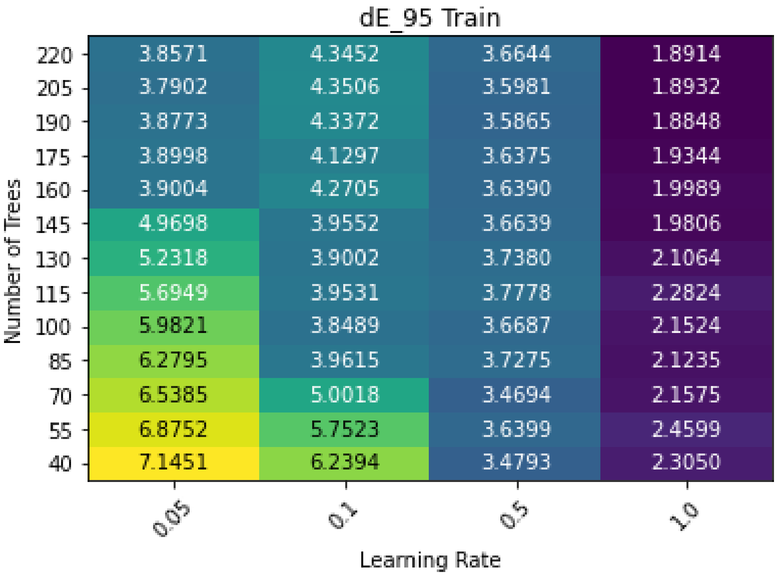
E.5.1 Train

In this sub-section, we provide the results obtained on the training data with different combinations of $learning_rate$ and $n\_estimators$.


MAE Train
Number of Trees
220 1.2459 1.0154 0.6248 0.5189
205 1.2970 1.0047 0.6195 0.5331
190 1.3453 1.0245 0.6014 0.5345
175 1.4074 1.0998 0.6055 0.5202
160 1.4221 1.1359 0.6276 0.5330
145 1.4974 1.1513 0.6583 0.5346
130 1.5859 1.2097 0.6559 0.5297
115 1.6144 1.2641 0.6915 0.5462
100 1.7136 1.3196 0.7407 0.5615
85 1.7792 1.4087 0.7543 0.5890
70 1.9682 1.4911 0.7569 0.5678
55 2.0954 1.6392 0.7963 0.6314
40 2.3110 1.7826 0.9134 0.6744
0.05
0.1
0.5
1.0
Learning Rate


MSE Train
Number of Trees
220 2.3223 1.5722 0.6407 0.4387
205 2.4819 1.5662 0.6307 0.4620
190 2.6368 1.6192 0.6001 0.4639
175 2.8419 1.8407 0.6023 0.4460
160 2.9226 1.9507 0.6467 0.4718
145 3.1974 1.9974 0.7023 0.4780
130 3.5202 2.1617 0.7035 0.4699
115 3.6373 2.3536 0.7921 0.4953
100 4.0423 2.5179 0.9074 0.5169
85 4.3608 2.8017 0.9419 0.5567
70 5.1648 3.1183 0.9650 0.5376
55 5.8533 3.7482 1.0662 0.6680
40 7.0486 4.3823 1.2951 0.7780
0.05
0.1
0.5
1.0
Learning Rate


dE_avg Train
Number of Trees
220
205
190
175
160
145
130
115
100
85
70
55
40
2.1991
1.9879
1.3657
1.0214
2.1983
2.0161
1.3549
1.0205
2.2657
2.0442
1.3586
1.0151
2.2700
2.0583
1.3773
1.0316
2.2500
2.1341
1.3877
1.0496
2.6083
2.1015
1.4275
1.0472
2.7473
2.1657
1.4456
1.0646
2.9091
2.2298
1.5086
1.1067
3.0920
2.2133
1.5051
1.1005
3.2983
2.3123
1.5912
1.0987
3.5367
2.6000
1.5231
1.1134
3.7532
2.9522
1.6639
1.2162
4.0476
3.3009
1.7263
1.1921
0.05
0.1
0.5
1.0
Learning Rate


dE_95 Train
Number of Trees
220
205
190
175
160
145
130
115
100
85
70
55
40
3.8571
4.3452
3.6644
1.8914
3.7902
4.3506
3.5981
1.8932
3.8773
4.3372
3.5865
1.8848
3.8998
4.1297
3.6375
1.9344
3.9004
4.2705
3.6390
1.9989
4.9698
3.9552
3.6639
1.9806
5.2318
3.9002
3.7380
2.1064
5.6949
3.9531
3.7778
2.2824
5.9821
3.8489
3.6687
2.1524
6.2795
3.9615
3.7275
2.1235
6.5385
5.0018
3.4694
2.1575
6.8752
5.7523
3.6399
2.4599
7.1451
6.2394
3.4793
2.3050
0.05
0.1
0.5
1.0
Learning Rate

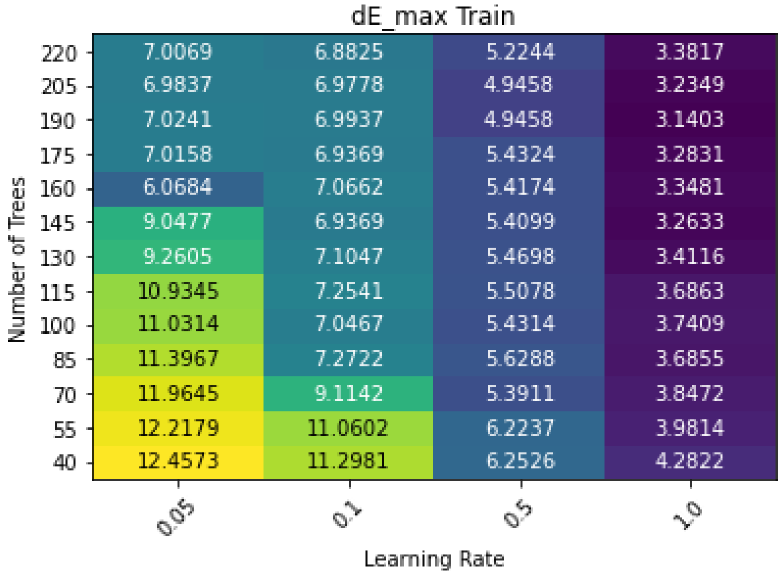
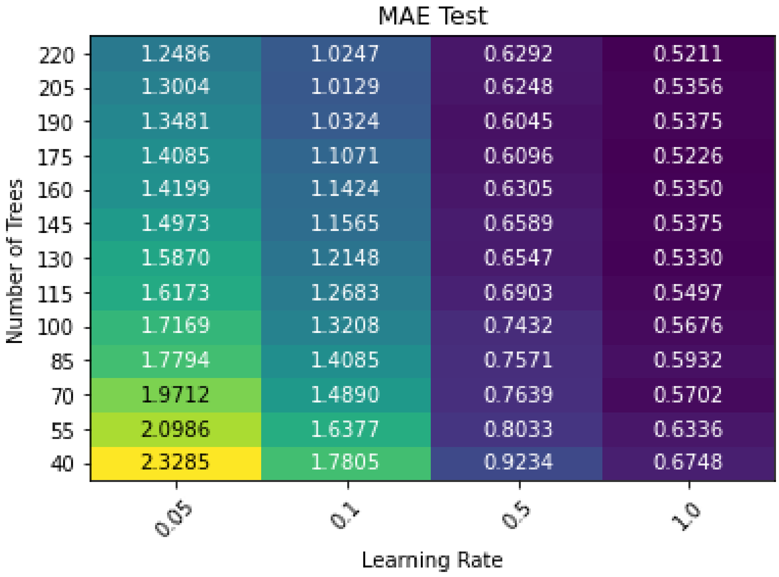
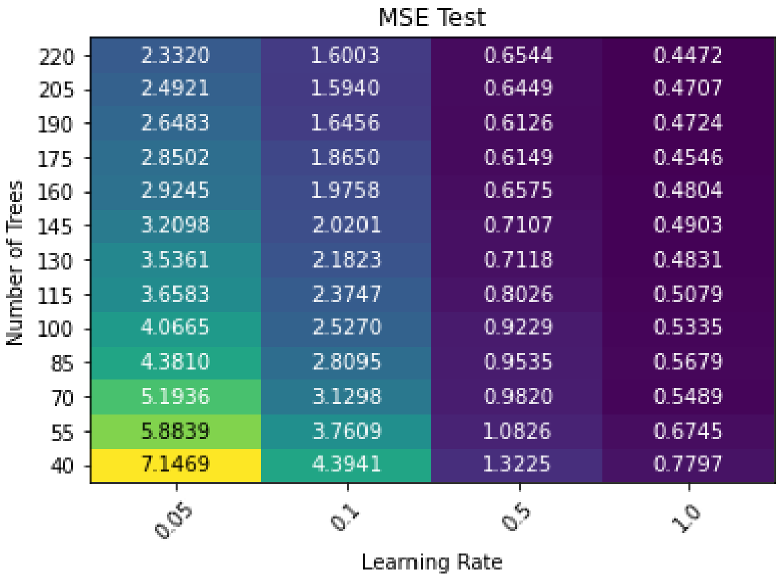


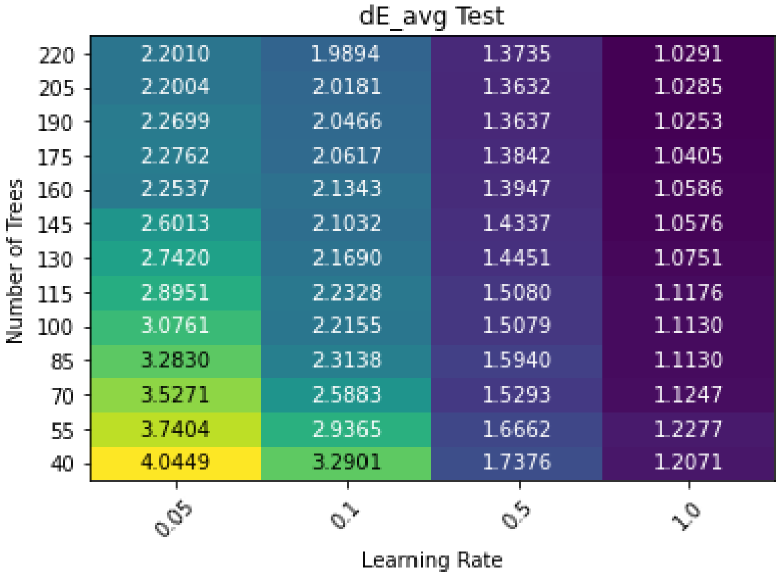
Figure E.10: Plot of Errors and δE on the Training Data for Adaboost

E.5.2 Test

In this sub-section, we provide the results obtained on the testing data with different combinations of $learning_{r}ate$ and $n\_estimators$.




MSE Test
Number of Trees
Learning Rate
0.05
0.1
0.5
1.0
220 2.3320 1.6003 0.6544 0.4472
205 2.4921 1.5940 0.6449 0.4707
190 2.6483 1.6456 0.6126 0.4724
175 2.8502 1.8650 0.6149 0.4546
160 2.9245 1.9758 0.6575 0.4804
145 3.2098 2.0201 0.7107 0.4903
130 3.5361 2.1823 0.7118 0.4831
115 3.6583 2.3747 0.8026 0.5079
100 4.0665 2.5270 0.9229 0.5335
85 4.3810 2.8095 0.9535 0.5679
70 5.1936 3.1298 0.9820 0.5489
55 5.8839 3.7609 1.0826 0.6745
40 7.1469 4.3941 1.3225 0.7797


dE_avg Test
Number of Trees
Learning Rate
0.05
0.1
0.5
1.0
220 2.2010 1.9894 1.3735 1.0291
205 2.2004 2.0181 1.3632 1.0285
190 2.2699 2.0466 1.3637 1.0253
175 2.2762 2.0617 1.3842 1.0405
160 2.2537 2.1343 1.3947 1.0586
145 2.6013 2.1032 1.4337 1.0576
130 2.7420 2.1690 1.4451 1.0751
115 2.8951 2.2328 1.5080 1.1176
100 3.0761 2.2155 1.5079 1.1130
85 3.2830 2.3138 1.5940 1.1130
70 3.5271 2.5883 1.5293 1.1247
55 3.7404 2.9365 1.6662 1.2277
40 4.0449 3.2901 1.7376 1.2071

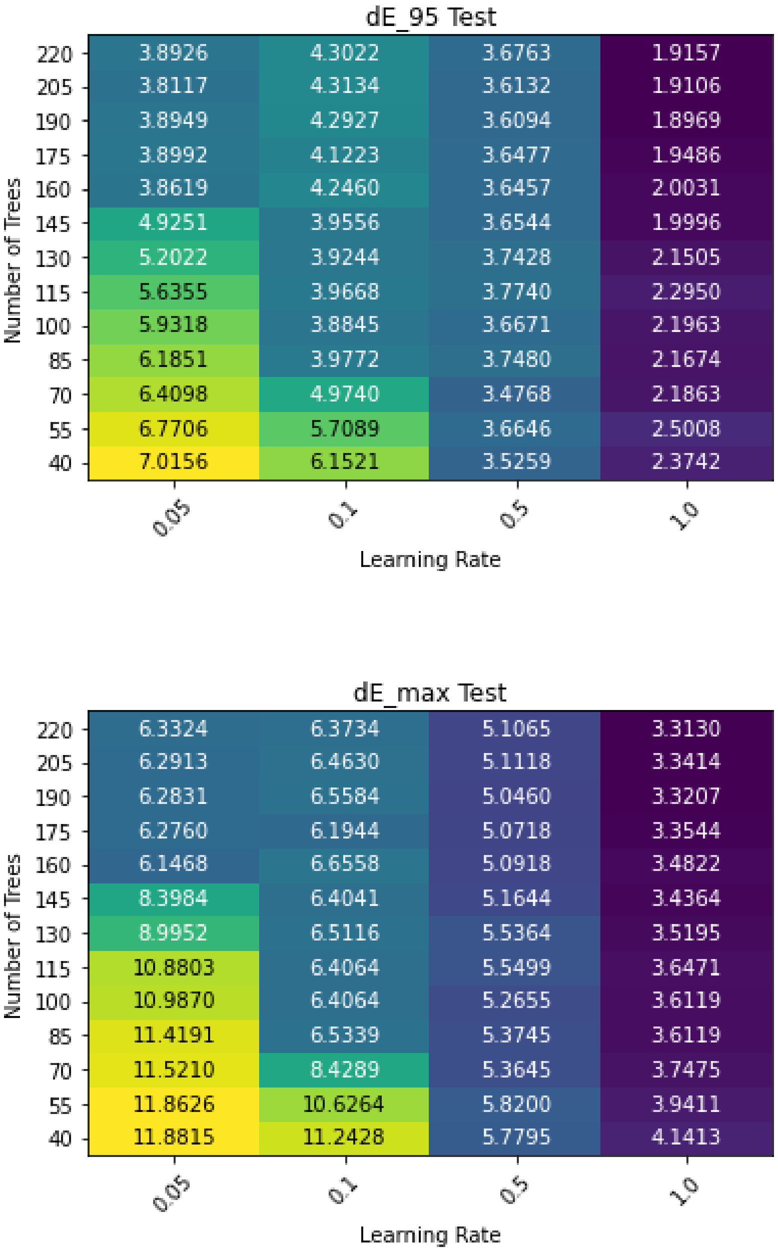


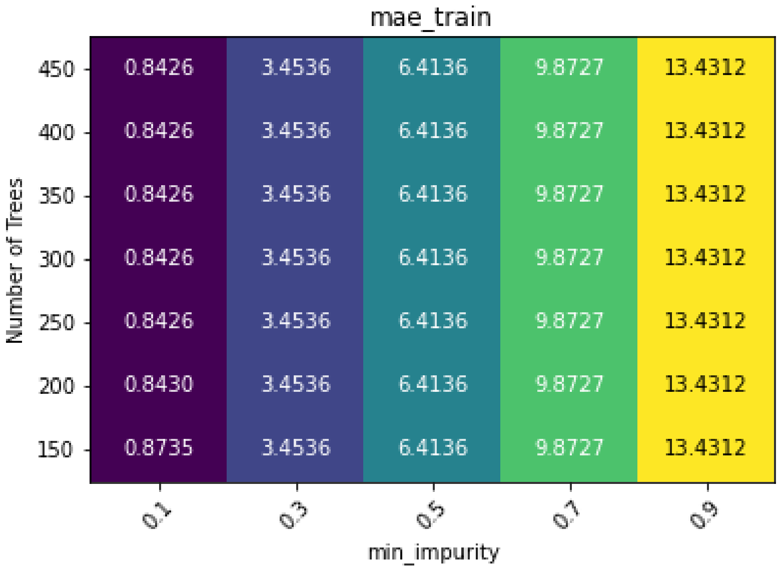
Figure E.11: Plot of Errors and δE on the Testing Data for Adaboost

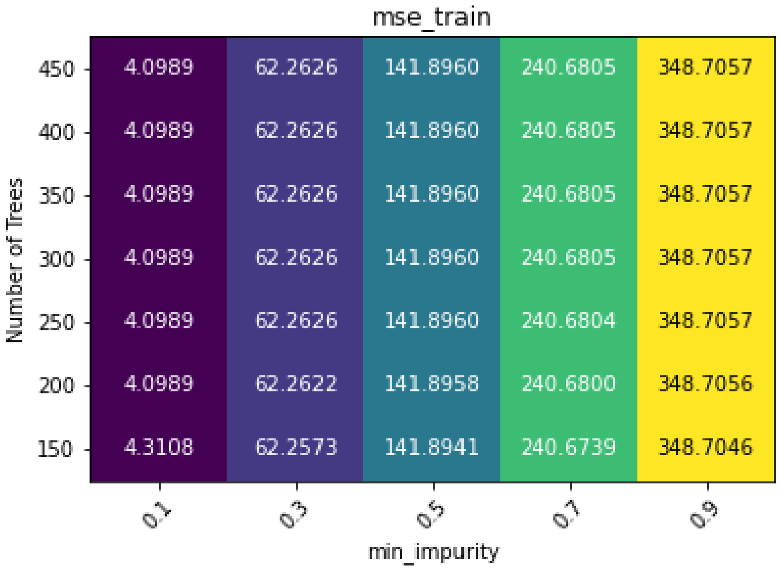
E.6 Gradient Boost (GB) Regression

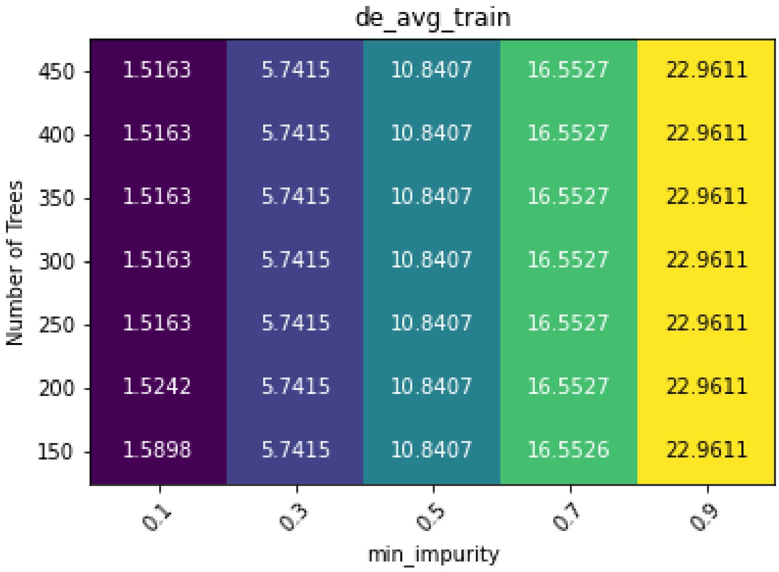
In this section, we provide the results obtained on performing grid search on Gradient Boost algorithm with different hyperparameters.

E.6.1 Train

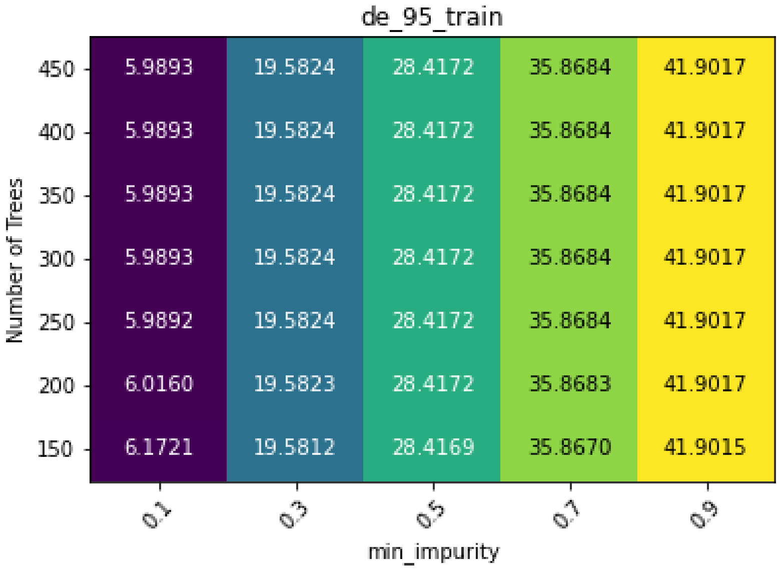
In this sub-section, we provide the results obtained on the training data with different combinations of *min_impurity* and *n_estimator*.


mae_train
Number of Trees
450 0.8426 3.4536 6.4136 9.8727 13.4312
400 0.8426 3.4536 6.4136 9.8727 13.4312
350 0.8426 3.4536 6.4136 9.8727 13.4312
300 0.8426 3.4536 6.4136 9.8727 13.4312
250 0.8426 3.4536 6.4136 9.8727 13.4312
200 0.8430 3.4536 6.4136 9.8727 13.4312
150 0.8735 3.4536 6.4136 9.8727 13.4312
0.1
0.3
0.5
0.7
0.9
min_impurity


mse_train
Number of Trees
450 4.0989 62.2626 141.8960 240.6805 348.7057
400 4.0989 62.2626 141.8960 240.6805 348.7057
350 4.0989 62.2626 141.8960 240.6805 348.7057
300 4.0989 62.2626 141.8960 240.6805 348.7057
250 4.0989 62.2626 141.8960 240.6804 348.7057
200 4.0989 62.2622 141.8958 240.6800 348.7056
150 4.3108 62.2573 141.8941 240.6739 348.7046
0.1
0.3
0.5
0.7
0.9
min_impurity



Number of Trees	0.1	0.3	0.5	0.7	0.9
450	1.5163	5.7415	10.8407	16.5527	22.9611
400	1.5163	5.7415	10.8407	16.5527	22.9611
350	1.5163	5.7415	10.8407	16.5527	22.9611
300	1.5163	5.7415	10.8407	16.5527	22.9611
250	1.5163	5.7415	10.8407	16.5527	22.9611
200	1.5242	5.7415	10.8407	16.5527	22.9611
150	1.5898	5.7415	10.8407	16.5526	22.9611



Number of Trees	0.1	0.3	0.5	0.7	0.9
450	5.9893	19.5824	28.4172	35.8684	41.9017
400	5.9893	19.5824	28.4172	35.8684	41.9017
350	5.9893	19.5824	28.4172	35.8684	41.9017
300	5.9893	19.5824	28.4172	35.8684	41.9017
250	5.9892	19.5824	28.4172	35.8684	41.9017
200	6.0160	19.5823	28.4172	35.8683	41.9017
150	6.1721	19.5812	28.4169	35.8670	41.9015

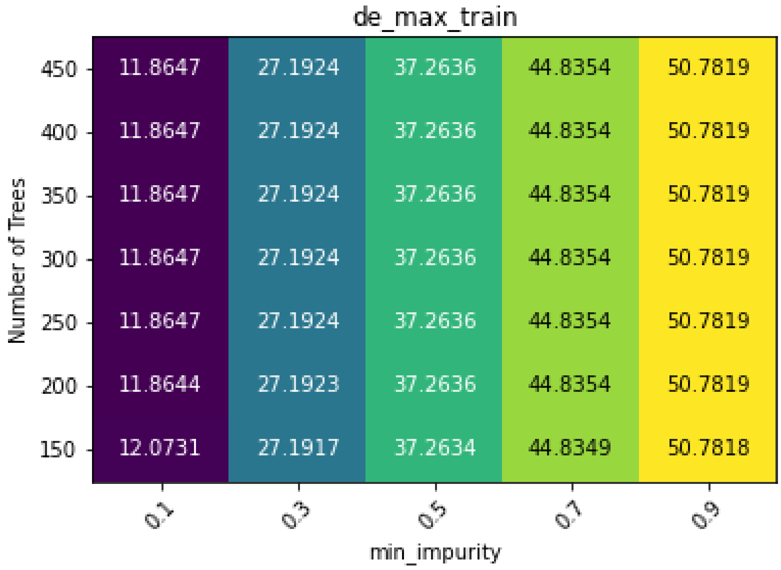
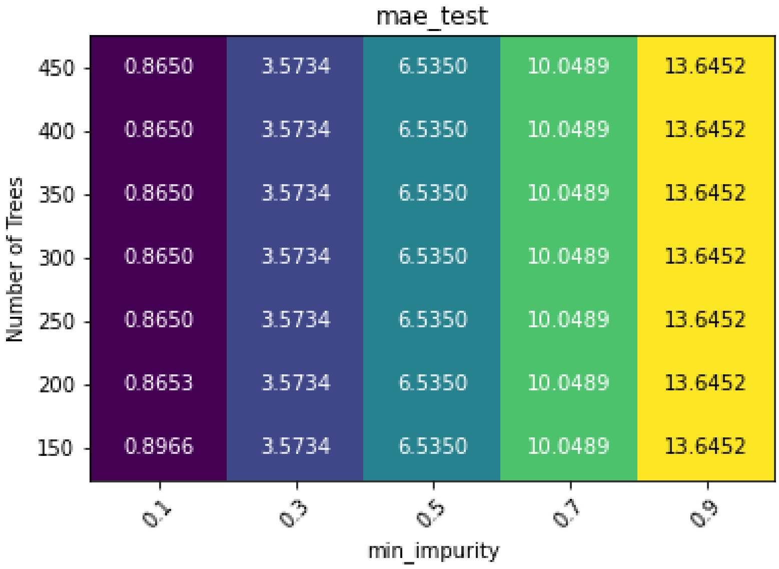
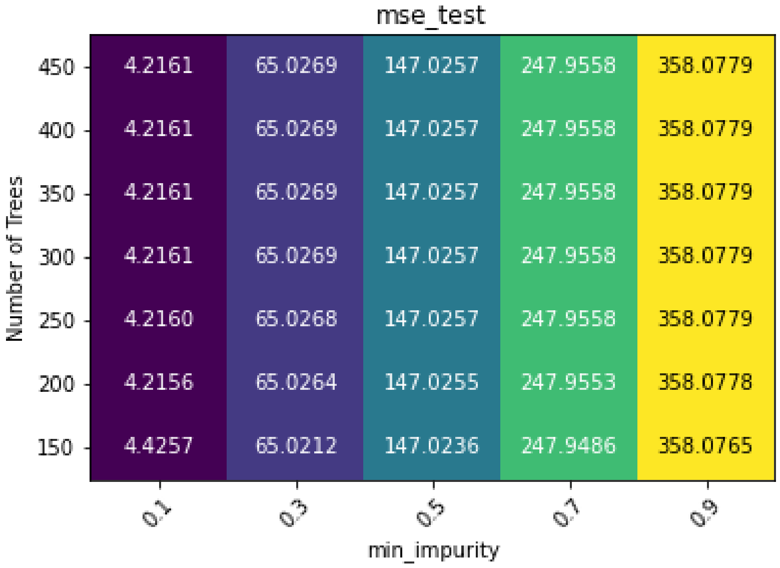


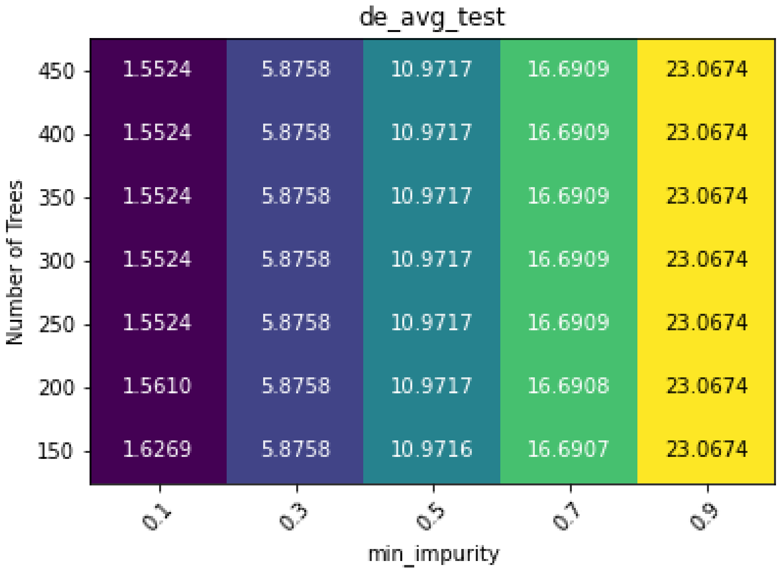
Figure E.12: Plot of Errors and δE on the Training Data

E.6.2 Test

In this sub-section, we provide the results obtained on the testing data with different combinations of *min_impurity* and *n_estimator*.




mse_test
Number of Trees
450
4.2161
65.0269
147.0257
247.9558
358.0779
400
4.2161
65.0269
147.0257
247.9558
358.0779
350
4.2161
65.0269
147.0257
247.9558
358.0779
300
4.2161
65.0269
147.0257
247.9558
358.0779
250
4.2160
65.0268
147.0257
247.9558
358.0779
200
4.2156
65.0264
147.0255
247.9553
358.0778
150
4.4257
65.0212
147.0236
247.9486
358.0765
0.1
0.3
0.5
0.7
0.9
min_impurity


de_avg_test
Number of Trees
450
1.5524
5.8758
10.9717
16.6909
23.0674
400
1.5524
5.8758
10.9717
16.6909
23.0674
350
1.5524
5.8758
10.9717
16.6909
23.0674
300
1.5524
5.8758
10.9717
16.6909
23.0674
250
1.5524
5.8758
10.9717
16.6909
23.0674
200
1.5610
5.8758
10.9717
16.6908
23.0674
150
1.6269
5.8758
10.9716
16.6907
23.0674
0.1
0.3
0.5
0.7
0.9
min_impurity

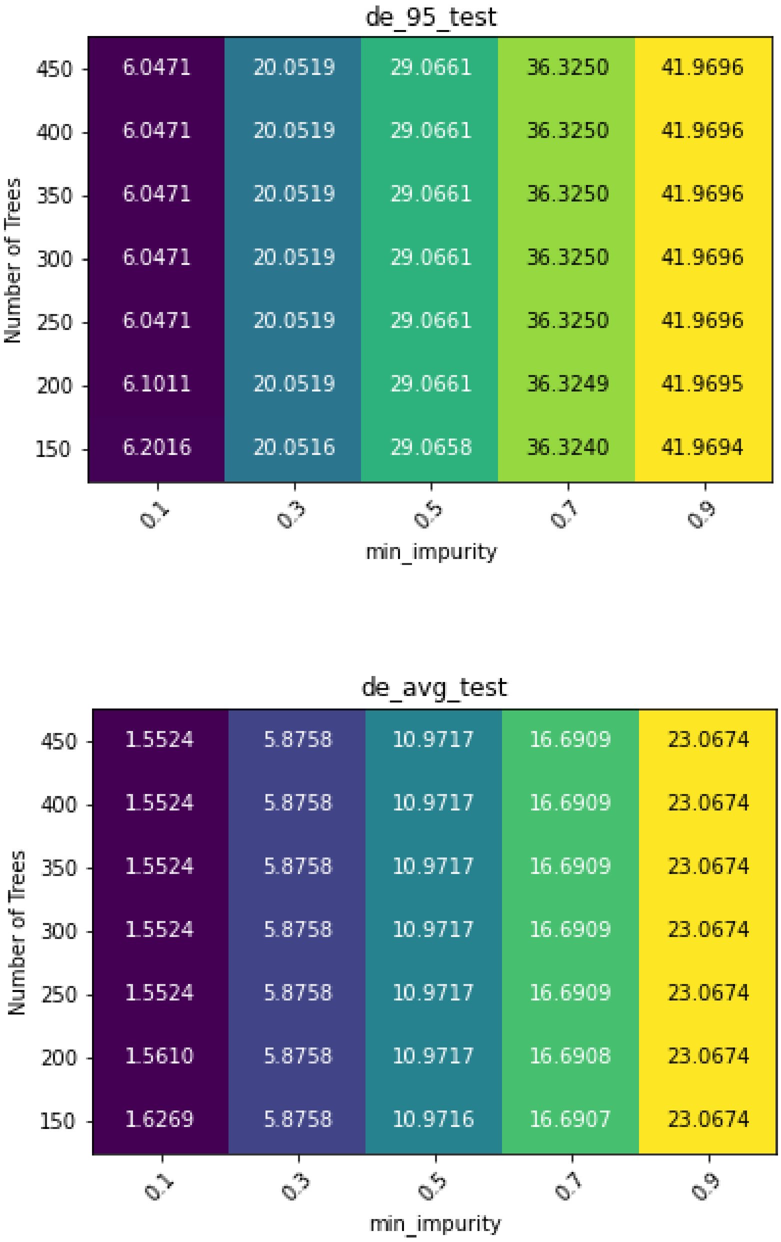


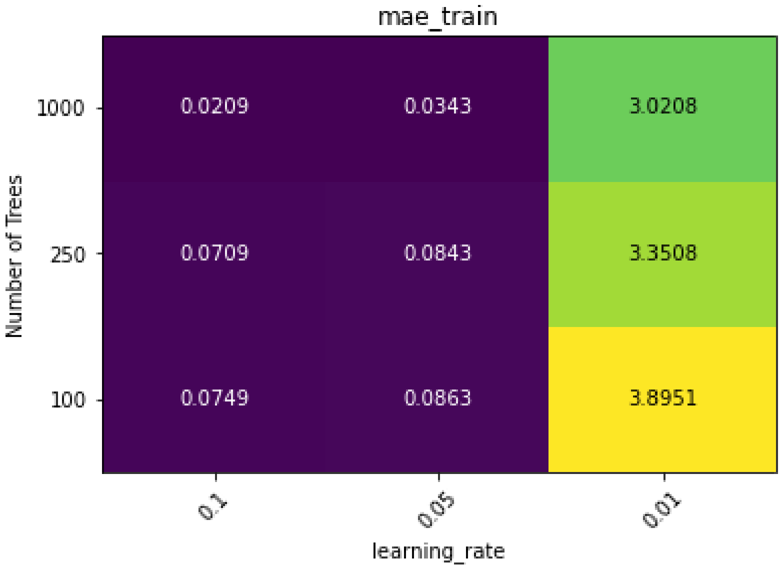
Figure E.13: Plot of Errors and δE on the Testing Data

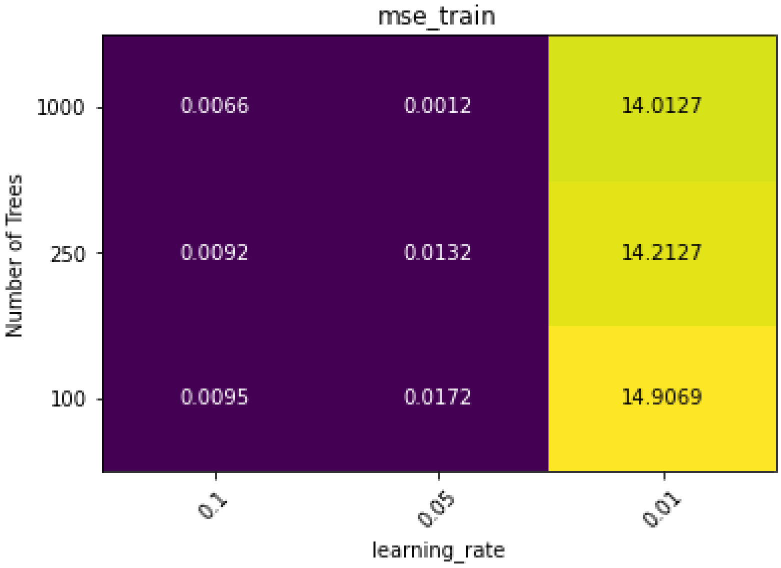
E.7 XGBoost (XGB) Regression

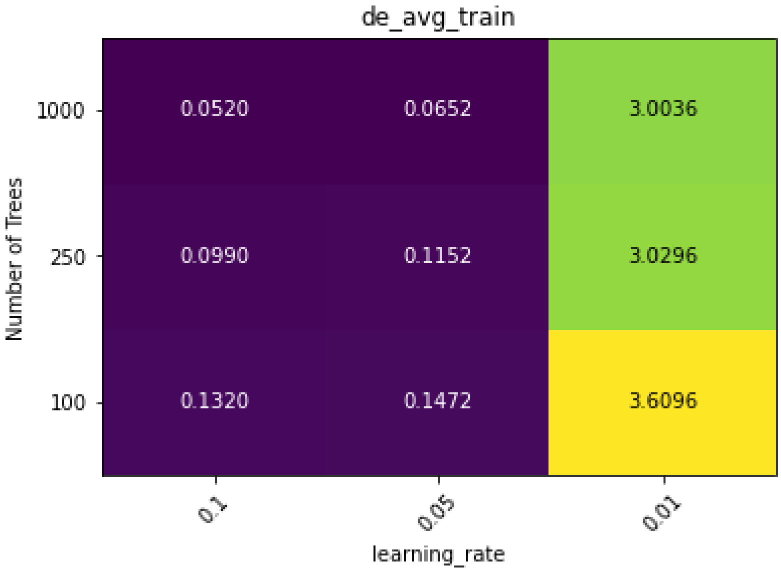
In this section, we provide the results obtained on performing grid search on XGBoost algorithm with different hyperparameters.

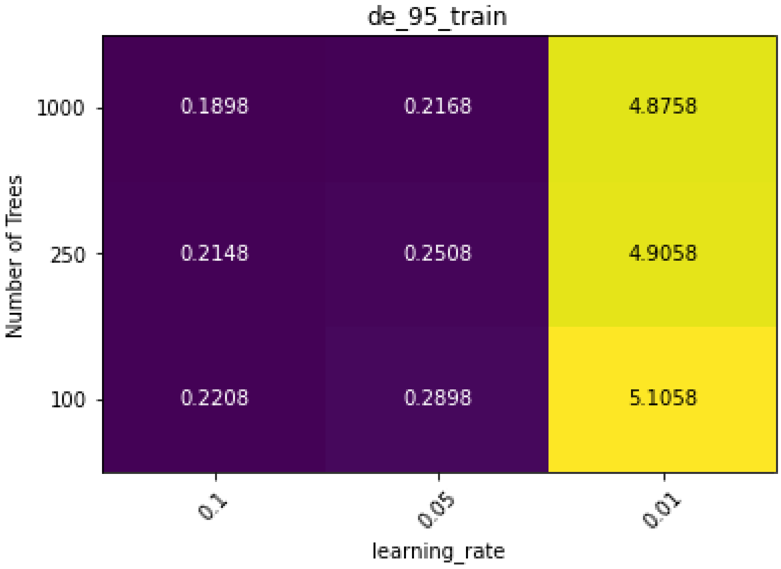
E.7.1 Train

In this sub-section, we provide the results obtained on the training data with different combinations of *learning_rate* and *n_estimator*.


mae_train
Number of Trees
1000
250
100
0.0209
0.0343
3.0208
0.0709
0.0843
3.3508
0.0749
0.0863
3.8951
0.1
0.05
0.01
learning_rate


mse_train
Number of Trees
1000
250
100
0.0066
0.0012
14.0127
0.0092
0.0132
14.2127
0.0095
0.0172
14.9069
0.1
0.05
0.01
learning_rate


de_avg_train
Number of Trees
1000
250
100
0.0520
0.0652
3.0036
0.0990
0.1152
3.0296
0.1320
0.1472
3.6096
0.1
0.05
0.01
learning_rate


de_95_train
Number of Trees
1000
250
100
0.1898
0.2168
4.8758
0.2148
0.2508
4.9058
0.2208
0.2898
5.1058
0.1
0.05
0.01
learning_rate

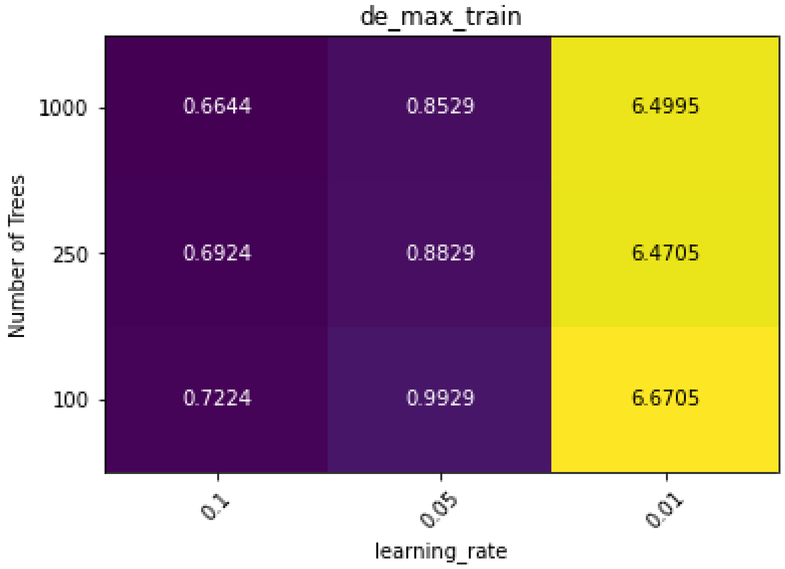
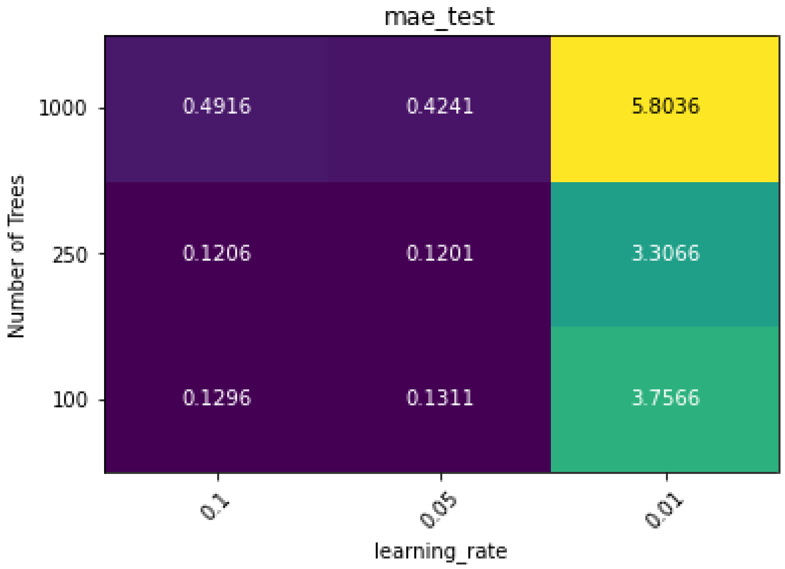
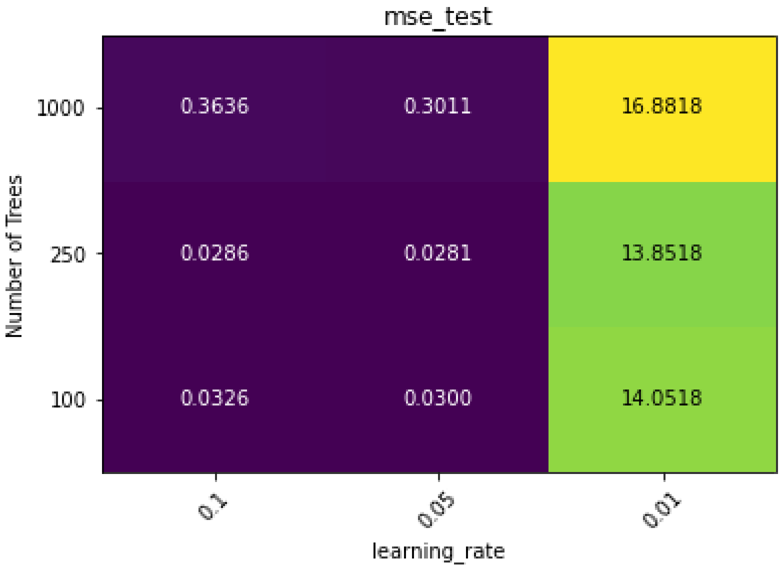


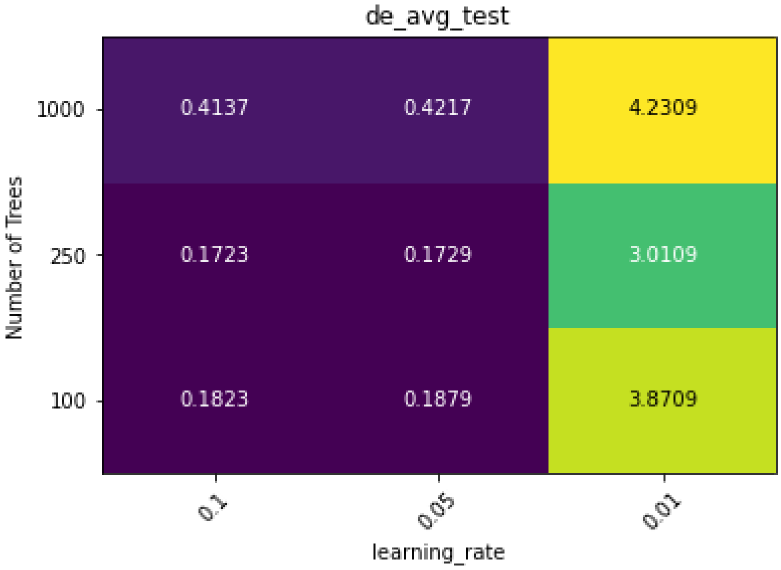
Figure E.14: Plot of Errors and δE on the Training Data for XGBoost

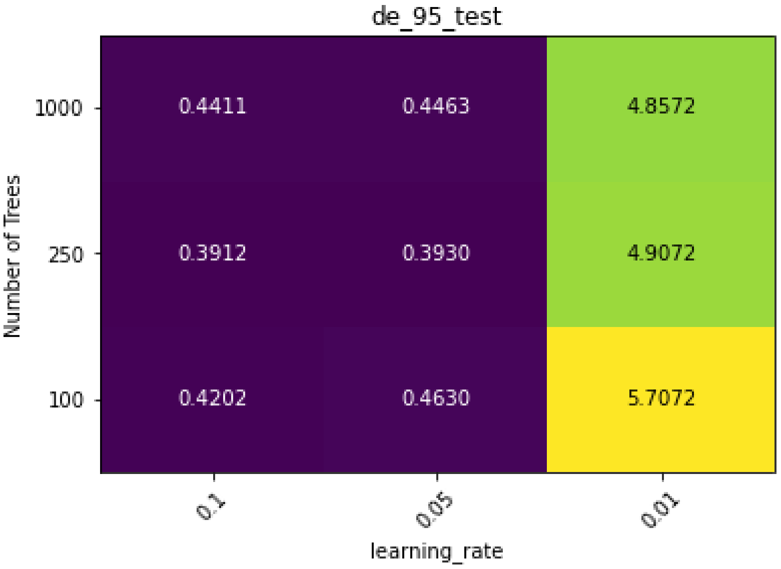
E.7.2 Test

In this sub-section, we provide the results obtained on the testing data with different combinations of *learning_rate* and *n_estimator*.




mse_test
Number of Trees
1000
250
100
0.3636
0.3011
16.8818
0.0286
0.0281
13.8518
0.0326
0.0300
14.0518
0.1
0.05
0.01
learning_rate


de_avg_test
Number of Trees
1000
250
100
0.4137
0.4217
4.2309
0.1723
0.1729
3.0109
0.1823
0.1879
3.8709
0.1
0.05
0.01
learning_rate



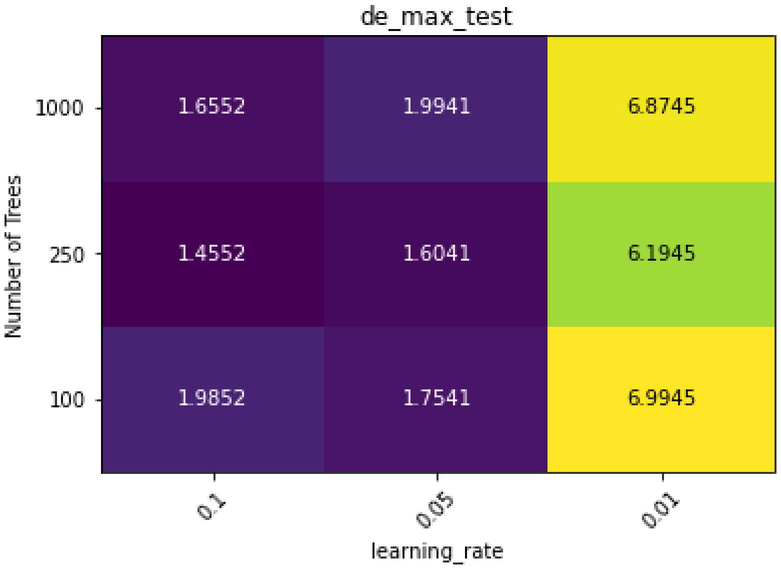


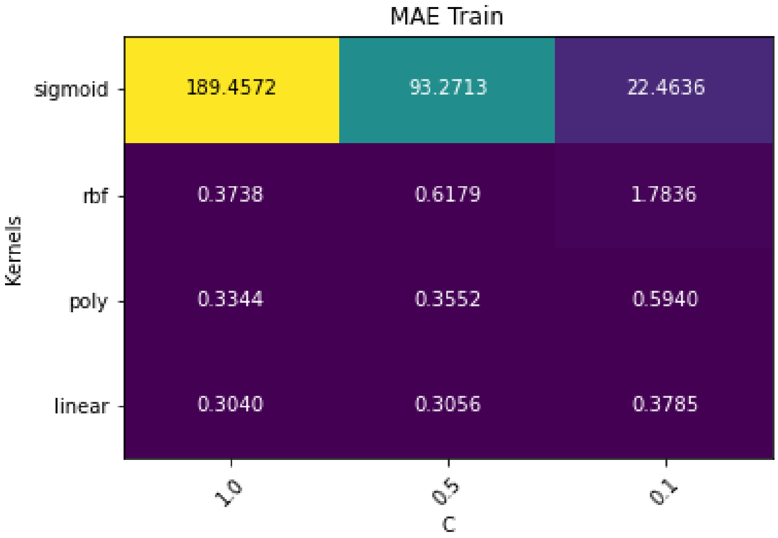
Figure E.15: Plot of Errors and δE on the Testing Data for XGBoost

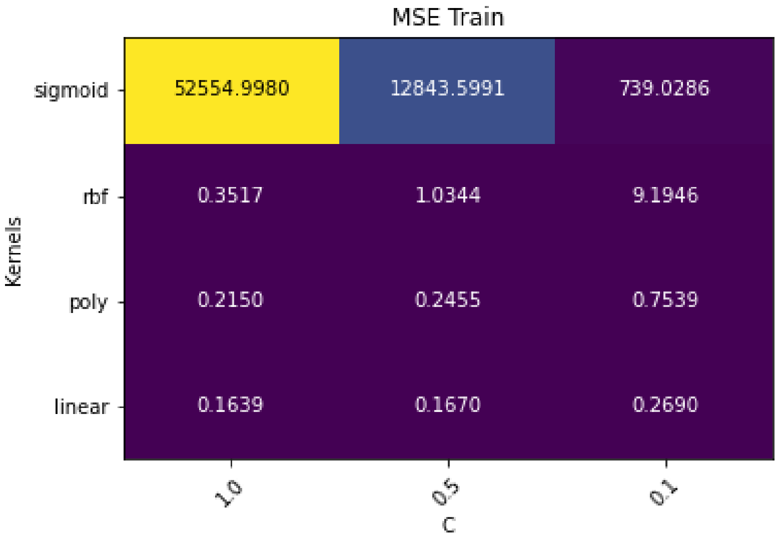
E.8 Support Vector (SVR) Regression

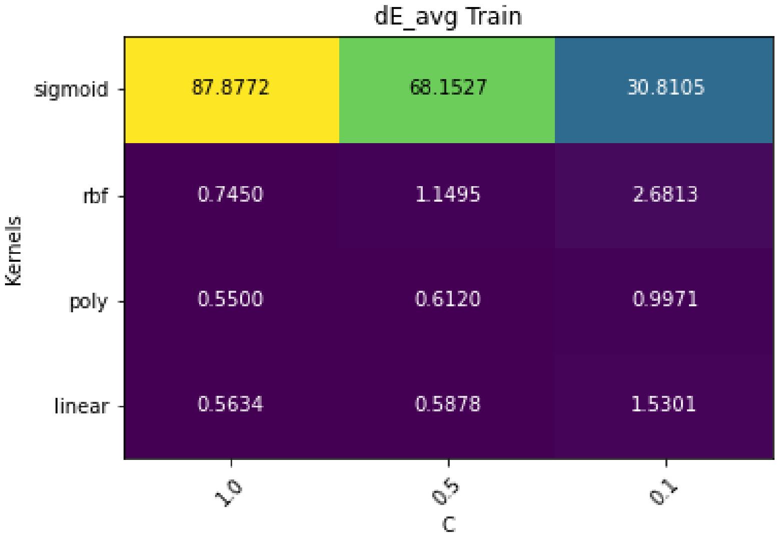
In this section, we provide the results obtained on performing grid search on Support Vector Regression algorithm with different hyperparameters.

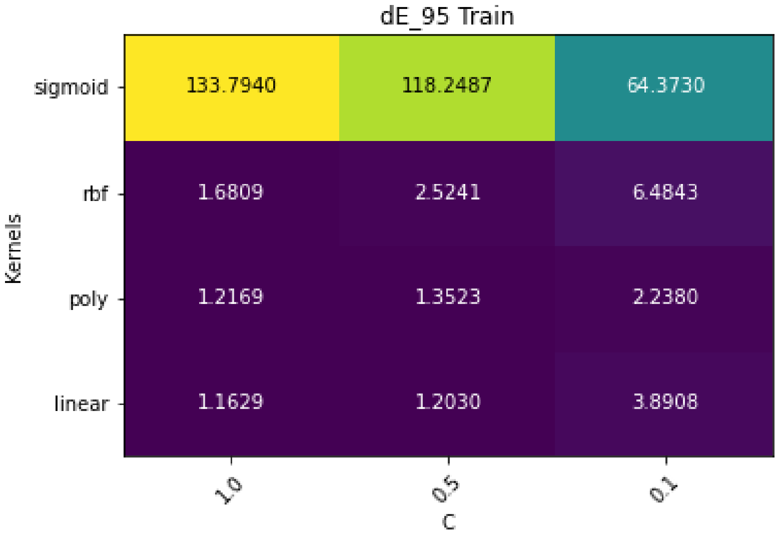
E.8.1 Train

In this sub-section, we provide the results obtained on the training data with different combinations of *kernels* and C.


MAE Train
sigmoid
rbf
poly
linear
Kernels
189.4572
93.2713
22.4636
0.3738
0.6179
1.7836
0.3344
0.3552
0.5940
0.3040
0.3056
0.3785
1.0
0.5
0.1
C


MSE Train
sigmoid
rbf
poly
linear
Kernels
52554.9980
12843.5991
739.0286
0.3517
1.0344
9.1946
0.2150
0.2455
0.7539
0.1639
0.1670
0.2690
1.0
0.5
0.1
C


dE_avg Train
sigmoid
rbf
poly
linear
Kernels
87.8772
68.1527
30.8105
0.7450
1.1495
2.6813
0.5500
0.6120
0.9971
0.5634
0.5878
1.5301
1.0
0.5
0.1
C


dE_95 Train
sigmoid
rbf
poly
linear
Kernels
133.7940
118.2487
64.3730
1.6809
2.5241
6.4843
1.2169
1.3523
2.2380
1.1629
1.2030
3.8908
1.0
0.5
0.1
C

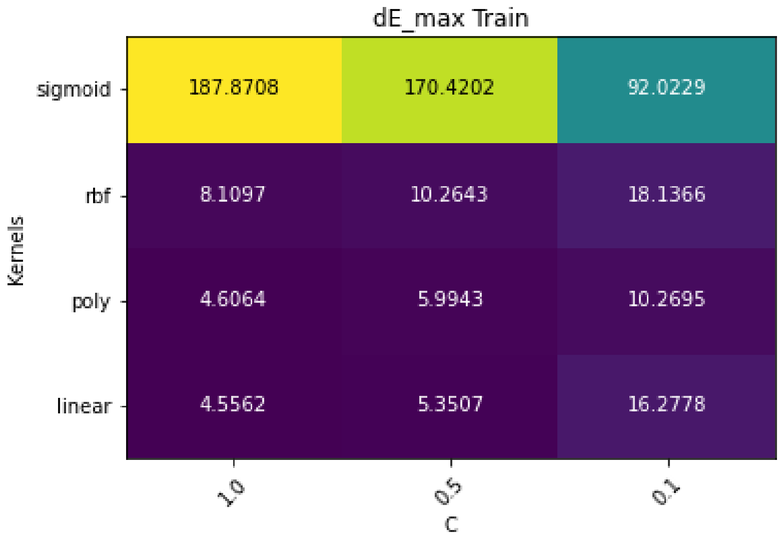
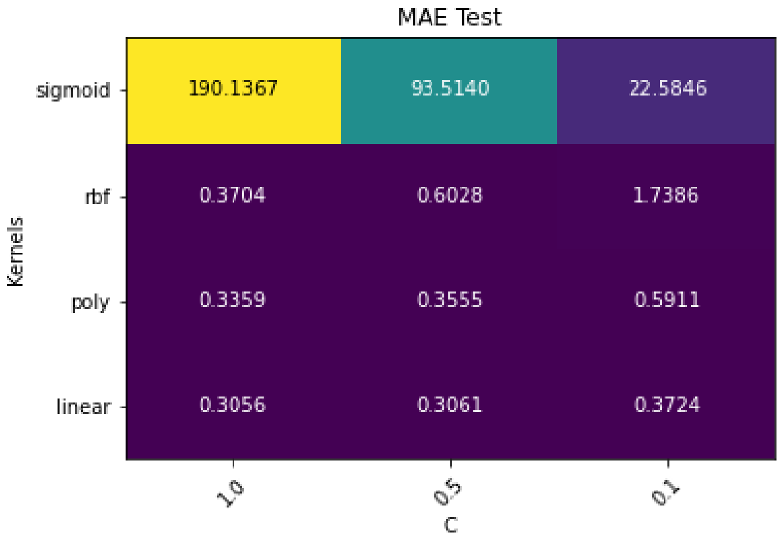
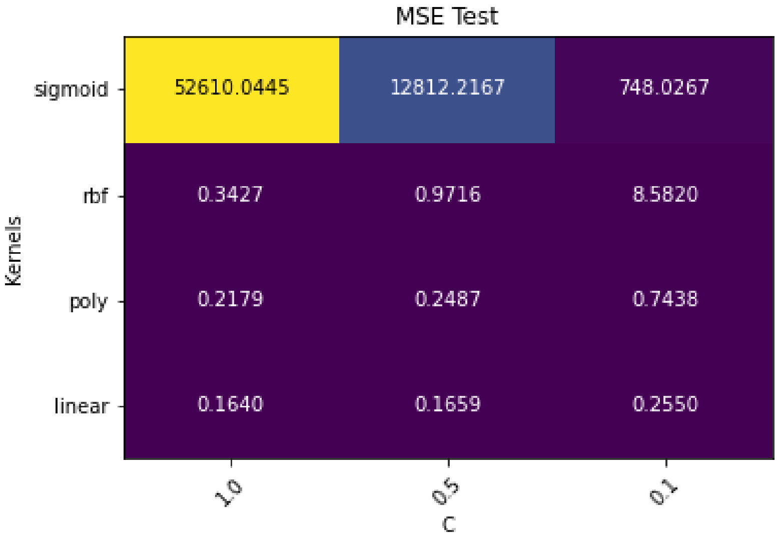


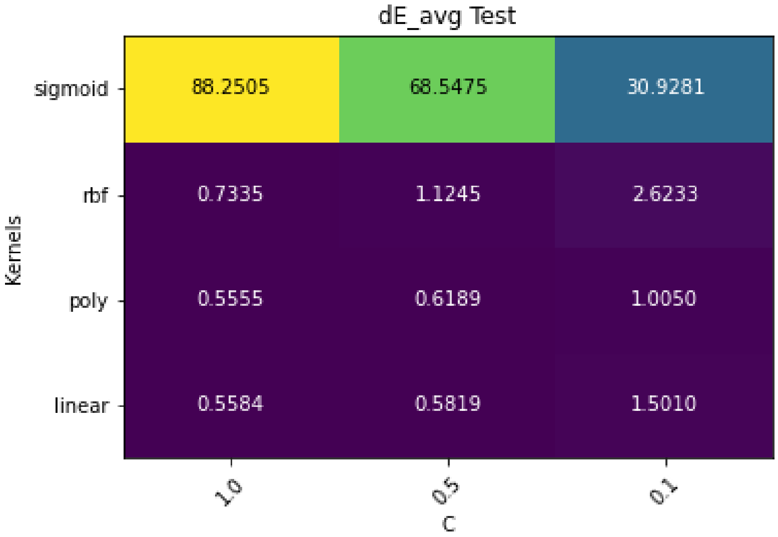
Figure E.16: Plot of Errors and δE on the Training Data

E.8.2 Test

In this sub-section, we provide the results obtained on the testing data with different combinations of *kernels* and C.




MSE Test
Kernels
sigmoid
rbf
poly
linear
52610.0445
12812.2167
748.0267
0.3427
0.9716
8.5820
0.2179
0.2487
0.7438
0.1640
0.1659
0.2550
1.0
0.5
0.1
C


dE_avg Test
Kernels
sigmoid
rbf
poly
linear
88.2505
68.5475
30.9281
0.7335
1.1245
2.6233
0.5555
0.6189
1.0050
0.5584
0.5819
1.5010
1.0
0.5
0.1
C

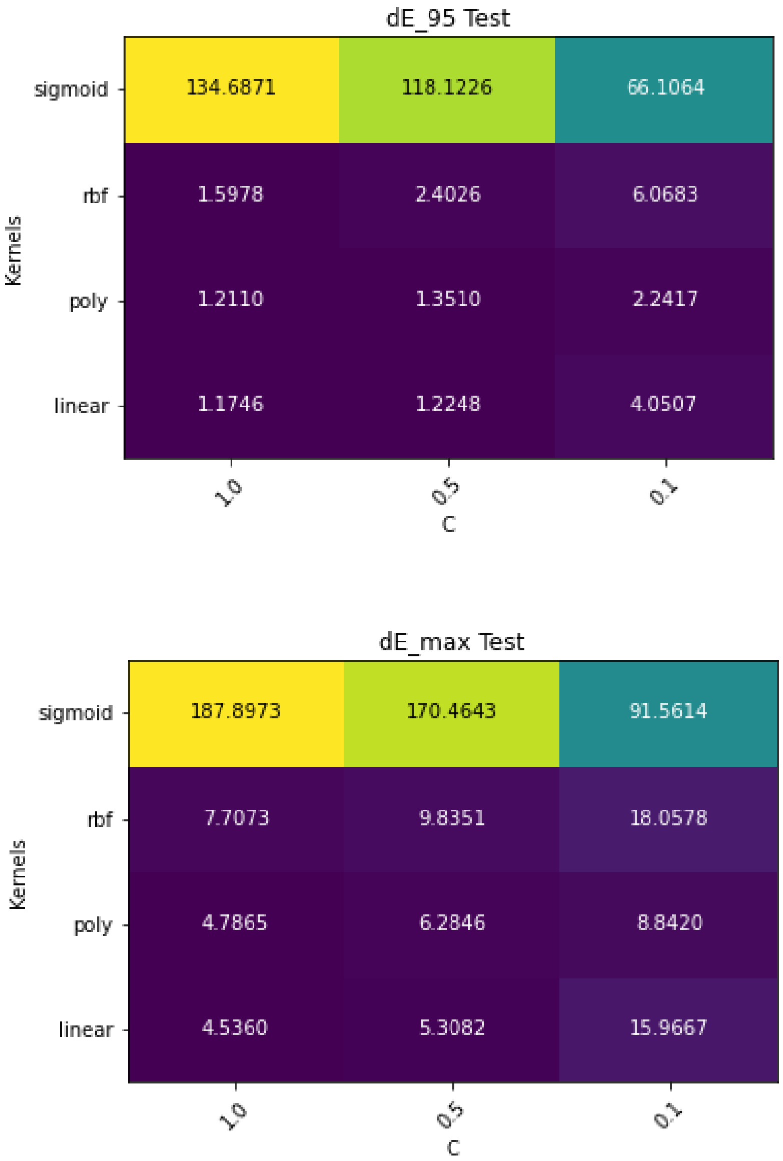


Figure E.17: Plot of Errors and δE on the Testing Data

Appendix F

Correlation with δE

In this chapter, we study the correlation between δE and all the features used in the regression models. Note that the δE is computed between Time 00 and Time 96. This analysis is different from the analyses we performed in the previous sections (i.e. the analysis of the correlations among all the features (Appendix D) and the feature importance analysis for different prediction models (Section 5.8.1) and (Section 6.7.2). In Appendix D, we discussed the correlation among the media characteristics, $L*$, $a*$, $b*$, C, M, Y and K. However, that analysis did not indicate the effect of each quantity on the change of color (measured by δE) upon drying. That analysis indicated on the correlation between each of the quantities. The feature importance analyses (Section 5.8.1 and Section 6.7.2) helped us to quantify the contribution of each of the feature for the prediction of $L*$, $a*$ and $b*$ values at Time 96. However, that analysis did not evaluate the effect of each of the features on the extent of drying (i.e. δE).

In this analysis, we study the contribution of each of the factors on δE. We know that correlation does not imply causation. However, we also know that the δE cannot have any effect on the features (or quantities) but the features have an effect on the δE. So, if we find high correlation between features and δE we can conclude that the feature has an effect on δE. The results of this analysis are as follows F.1.

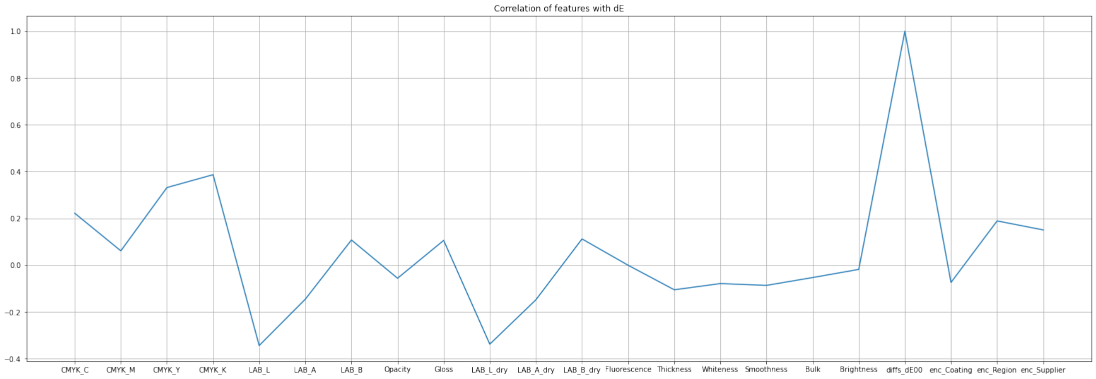


Figure F.1: Plot of Correlation between δE and all the features

In the above plot, we observe that the δE has a high correlation with Y and K. Generally speaking, the plot suggests that the color itself (represented by C, M, Y and K) contributes the most to the change of color. The $L*$, $a*$ and $b*$ values seem to have negative correlation with δE. Moreover, (most of) the media characteristics seem to have very small negative correlation with δE. However, the correlation of the media characteristics cannot be compared with C, M, Y and K (and $L*$, $a*$ and $b*$ values) due to insufficient data.

Appendix G

Test Chart Data

The following test chart is used to collect the data for this project.

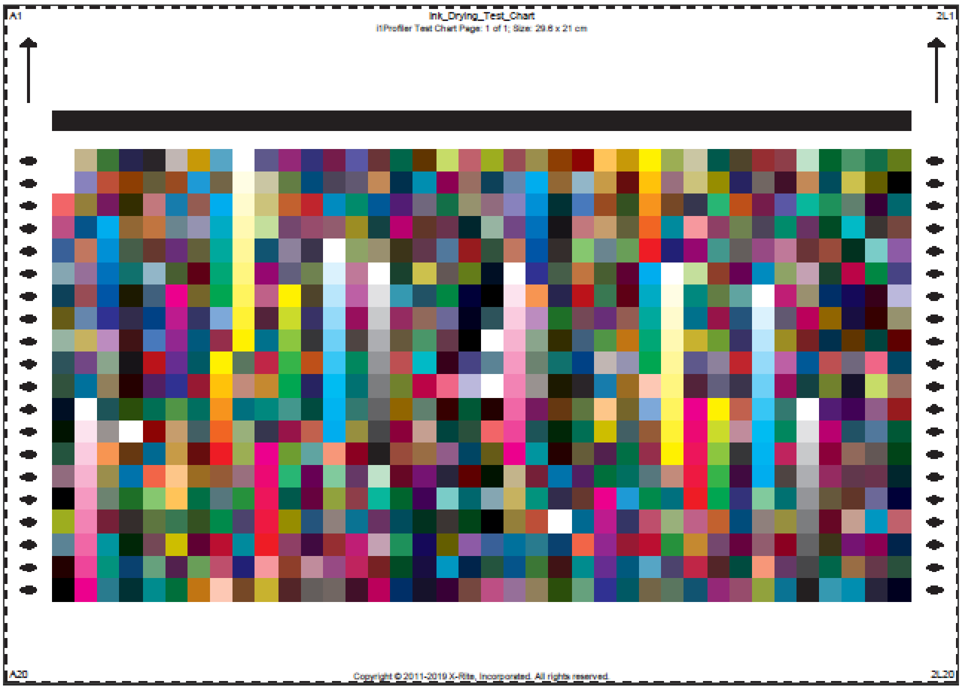
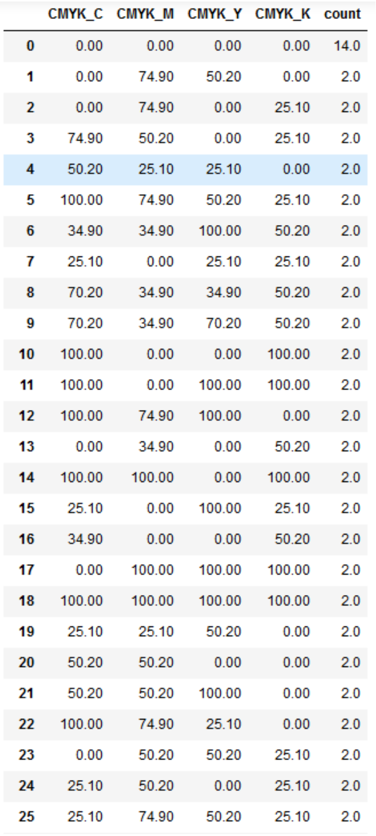
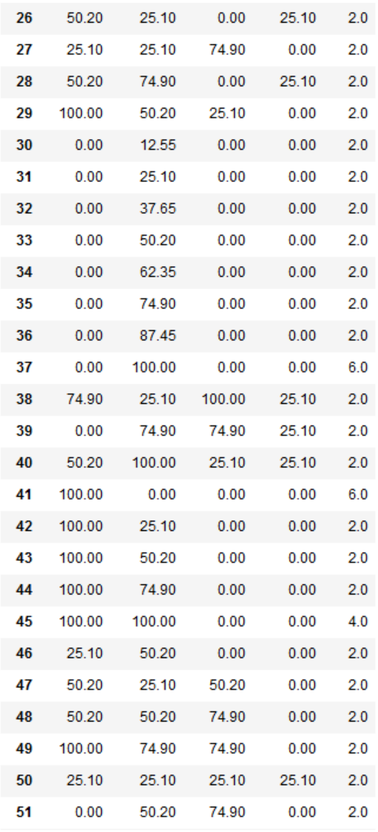

A1
Ink_Drying_Test_Chart
i1Profiler Test Chart Page: 1 of 1; Size: 29.6 x 21 cm
2L1
A20
Copyright © 2011-2019 X-Rite, Incorporated. All rights reserved.
2L20

Figure G.1: Color Test Chart

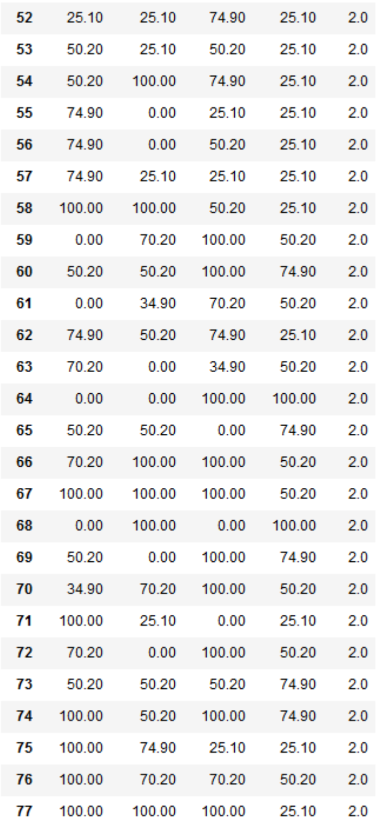
Following is the list of the colors (in CYMK format) in the test chart alongwith their frequencies.



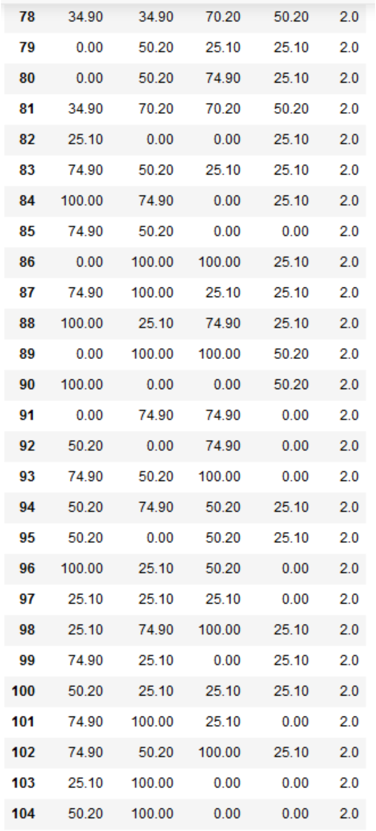
	CMYK_C	CMYK_M	CMYK_Y	CMYK_K	count
0	0.00	0.00	0.00	0.00	14.0
1	0.00	74.90	50.20	0.00	2.0
2	0.00	74.90	0.00	25.10	2.0
3	74.90	50.20	0.00	25.10	2.0
4	50.20	25.10	25.10	0.00	2.0
5	100.00	74.90	50.20	25.10	2.0
6	34.90	34.90	100.00	50.20	2.0
7	25.10	0.00	25.10	25.10	2.0
8	70.20	34.90	34.90	50.20	2.0
9	70.20	34.90	70.20	50.20	2.0
10	100.00	0.00	0.00	100.00	2.0
11	100.00	0.00	100.00	100.00	2.0
12	100.00	74.90	100.00	0.00	2.0
13	0.00	34.90	0.00	50.20	2.0
14	100.00	100.00	0.00	100.00	2.0
15	25.10	0.00	100.00	25.10	2.0
16	34.90	0.00	0.00	50.20	2.0
17	0.00	100.00	100.00	100.00	2.0
18	100.00	100.00	100.00	100.00	2.0
19	25.10	25.10	50.20	0.00	2.0
20	50.20	50.20	0.00	0.00	2.0
21	50.20	50.20	100.00	0.00	2.0
22	100.00	74.90	25.10	0.00	2.0
23	0.00	50.20	50.20	25.10	2.0
24	25.10	50.20	0.00	25.10	2.0
25	25.10	74.90	50.20	25.10	2.0



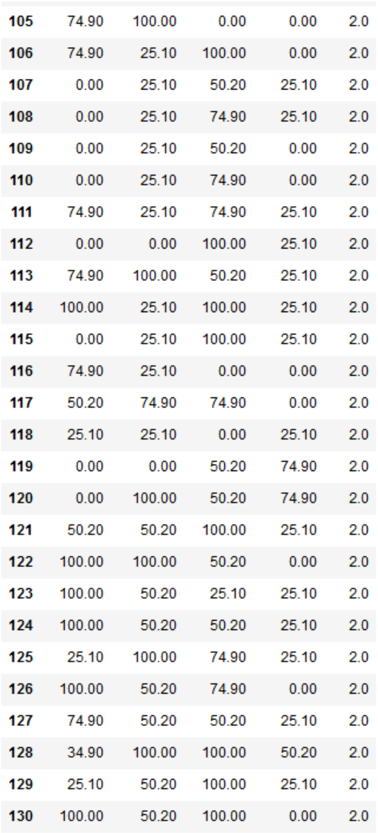
26	50.20	25.10	0.00	25.10	2.0
27	25.10	25.10	74.90	0.00	2.0
28	50.20	74.90	0.00	25.10	2.0
29	100.00	50.20	25.10	0.00	2.0
30	0.00	12.55	0.00	0.00	2.0
31	0.00	25.10	0.00	0.00	2.0
32	0.00	37.65	0.00	0.00	2.0
33	0.00	50.20	0.00	0.00	2.0
34	0.00	62.35	0.00	0.00	2.0
35	0.00	74.90	0.00	0.00	2.0
36	0.00	87.45	0.00	0.00	2.0
37	0.00	100.00	0.00	0.00	6.0
38	74.90	25.10	100.00	25.10	2.0
39	0.00	74.90	74.90	25.10	2.0
40	50.20	100.00	25.10	25.10	2.0
41	100.00	0.00	0.00	0.00	6.0
42	100.00	25.10	0.00	0.00	2.0
43	100.00	50.20	0.00	0.00	2.0
44	100.00	74.90	0.00	0.00	2.0
45	100.00	100.00	0.00	0.00	4.0
46	25.10	50.20	0.00	0.00	2.0
47	50.20	25.10	50.20	0.00	2.0
48	50.20	50.20	74.90	0.00	2.0
49	100.00	74.90	74.90	0.00	2.0
50	25.10	25.10	25.10	25.10	2.0
51	0.00	50.20	74.90	0.00	2.0



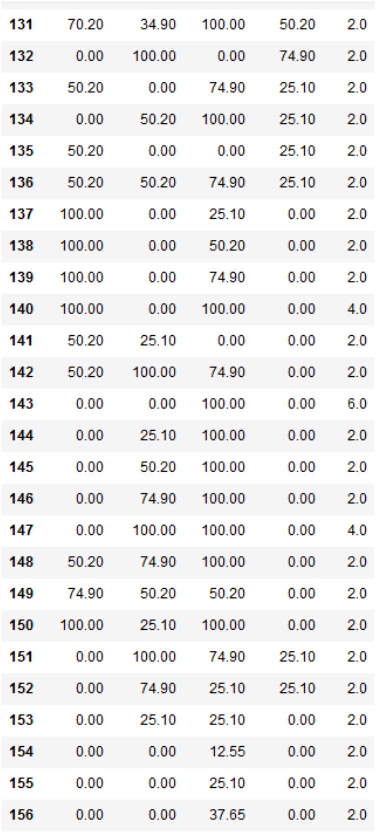
52	25.10	25.10	74.90	25.10	2.0
53	50.20	25.10	50.20	25.10	2.0
54	50.20	100.00	74.90	25.10	2.0
55	74.90	0.00	25.10	25.10	2.0
56	74.90	0.00	50.20	25.10	2.0
57	74.90	25.10	25.10	25.10	2.0
58	100.00	100.00	50.20	25.10	2.0
59	0.00	70.20	100.00	50.20	2.0
60	50.20	50.20	100.00	74.90	2.0
61	0.00	34.90	70.20	50.20	2.0
62	74.90	50.20	74.90	25.10	2.0
63	70.20	0.00	34.90	50.20	2.0
64	0.00	0.00	100.00	100.00	2.0
65	50.20	50.20	0.00	74.90	2.0
66	70.20	100.00	100.00	50.20	2.0
67	100.00	100.00	100.00	50.20	2.0
68	0.00	100.00	0.00	100.00	2.0
69	50.20	0.00	100.00	74.90	2.0
70	34.90	70.20	100.00	50.20	2.0
71	100.00	25.10	0.00	25.10	2.0
72	70.20	0.00	100.00	50.20	2.0
73	50.20	50.20	50.20	74.90	2.0
74	100.00	50.20	100.00	74.90	2.0
75	100.00	74.90	25.10	25.10	2.0
76	100.00	70.20	70.20	50.20	2.0
77	100.00	100.00	100.00	25.10	2.0



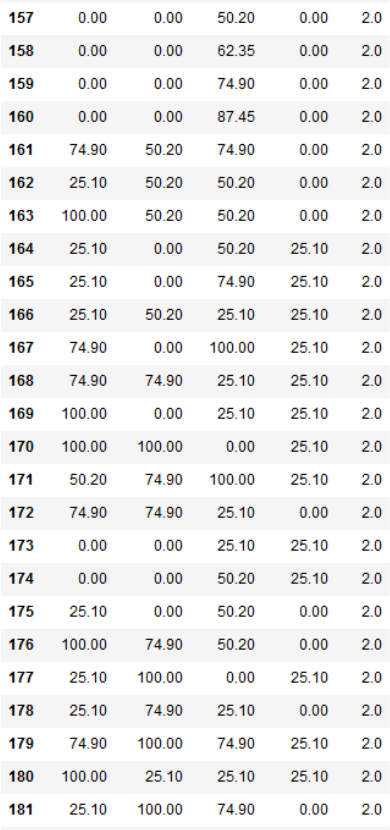
78	34.90	34.90	70.20	50.20	2.0
79	0.00	50.20	25.10	25.10	2.0
80	0.00	50.20	74.90	25.10	2.0
81	34.90	70.20	70.20	50.20	2.0
82	25.10	0.00	0.00	25.10	2.0
83	74.90	50.20	25.10	25.10	2.0
84	100.00	74.90	0.00	25.10	2.0
85	74.90	50.20	0.00	0.00	2.0
86	0.00	100.00	100.00	25.10	2.0
87	74.90	100.00	25.10	25.10	2.0
88	100.00	25.10	74.90	25.10	2.0
89	0.00	100.00	100.00	50.20	2.0
90	100.00	0.00	0.00	50.20	2.0
91	0.00	74.90	74.90	0.00	2.0
92	50.20	0.00	74.90	0.00	2.0
93	74.90	50.20	100.00	0.00	2.0
94	50.20	74.90	50.20	25.10	2.0
95	50.20	0.00	50.20	25.10	2.0
96	100.00	25.10	50.20	0.00	2.0
97	25.10	25.10	25.10	0.00	2.0
98	25.10	74.90	100.00	25.10	2.0
99	74.90	25.10	0.00	25.10	2.0
100	50.20	25.10	25.10	25.10	2.0
101	74.90	100.00	25.10	0.00	2.0
102	74.90	50.20	100.00	25.10	2.0
103	25.10	100.00	0.00	0.00	2.0
104	50.20	100.00	0.00	0.00	2.0



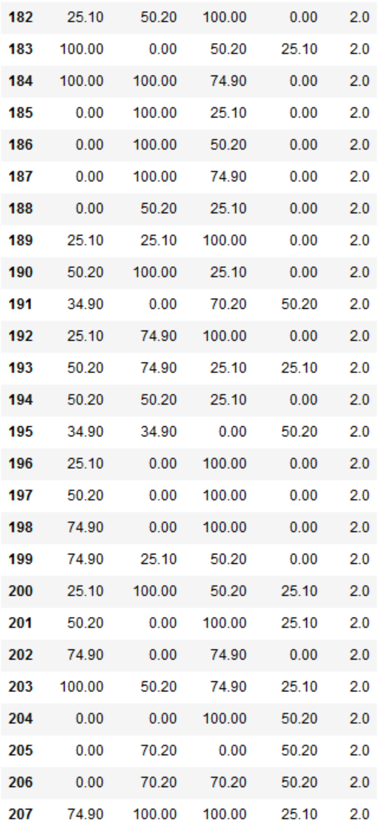
105	74.90	100.00	0.00	0.00	2.0
106	74.90	25.10	100.00	0.00	2.0
107	0.00	25.10	50.20	25.10	2.0
108	0.00	25.10	74.90	25.10	2.0
109	0.00	25.10	50.20	0.00	2.0
110	0.00	25.10	74.90	0.00	2.0
111	74.90	25.10	74.90	25.10	2.0
112	0.00	0.00	100.00	25.10	2.0
113	74.90	100.00	50.20	25.10	2.0
114	100.00	25.10	100.00	25.10	2.0
115	0.00	25.10	100.00	25.10	2.0
116	74.90	25.10	0.00	0.00	2.0
117	50.20	74.90	74.90	0.00	2.0
118	25.10	25.10	0.00	25.10	2.0
119	0.00	0.00	50.20	74.90	2.0
120	0.00	100.00	50.20	74.90	2.0
121	50.20	50.20	100.00	25.10	2.0
122	100.00	100.00	50.20	0.00	2.0
123	100.00	50.20	25.10	25.10	2.0
124	100.00	50.20	50.20	25.10	2.0
125	25.10	100.00	74.90	25.10	2.0
126	100.00	50.20	74.90	0.00	2.0
127	74.90	50.20	50.20	25.10	2.0
128	34.90	100.00	100.00	50.20	2.0
129	25.10	50.20	100.00	25.10	2.0
130	100.00	50.20	100.00	0.00	2.0



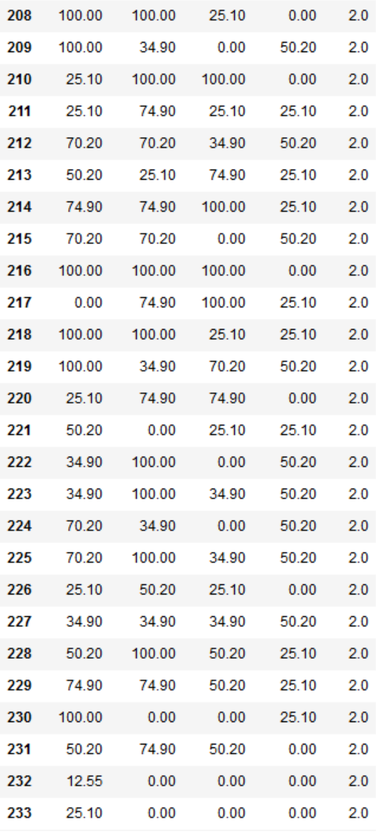
131	70.20	34.90	100.00	50.20	2.0
132	0.00	100.00	0.00	74.90	2.0
133	50.20	0.00	74.90	25.10	2.0
134	0.00	50.20	100.00	25.10	2.0
135	50.20	0.00	0.00	25.10	2.0
136	50.20	50.20	74.90	25.10	2.0
137	100.00	0.00	25.10	0.00	2.0
138	100.00	0.00	50.20	0.00	2.0
139	100.00	0.00	74.90	0.00	2.0
140	100.00	0.00	100.00	0.00	4.0
141	50.20	25.10	0.00	0.00	2.0
142	50.20	100.00	74.90	0.00	2.0
143	0.00	0.00	100.00	0.00	6.0
144	0.00	25.10	100.00	0.00	2.0
145	0.00	50.20	100.00	0.00	2.0
146	0.00	74.90	100.00	0.00	2.0
147	0.00	100.00	100.00	0.00	4.0
148	50.20	74.90	100.00	0.00	2.0
149	74.90	50.20	50.20	0.00	2.0
150	100.00	25.10	100.00	0.00	2.0
151	0.00	100.00	74.90	25.10	2.0
152	0.00	74.90	25.10	25.10	2.0
153	0.00	25.10	25.10	0.00	2.0
154	0.00	0.00	12.55	0.00	2.0
155	0.00	0.00	25.10	0.00	2.0
156	0.00	0.00	37.65	0.00	2.0



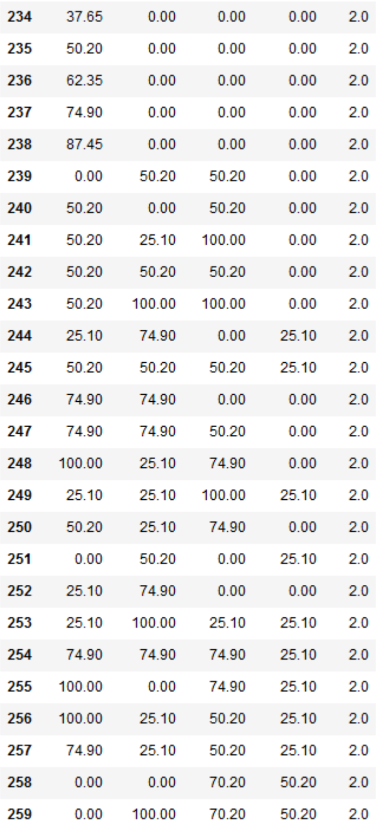
157	0.00	0.00	50.20	0.00	2.0
158	0.00	0.00	62.35	0.00	2.0
159	0.00	0.00	74.90	0.00	2.0
160	0.00	0.00	87.45	0.00	2.0
161	74.90	50.20	74.90	0.00	2.0
162	25.10	50.20	50.20	0.00	2.0
163	100.00	50.20	50.20	0.00	2.0
164	25.10	0.00	50.20	25.10	2.0
165	25.10	0.00	74.90	25.10	2.0
166	25.10	50.20	25.10	25.10	2.0
167	74.90	0.00	100.00	25.10	2.0
168	74.90	74.90	25.10	25.10	2.0
169	100.00	0.00	25.10	25.10	2.0
170	100.00	100.00	0.00	25.10	2.0
171	50.20	74.90	100.00	25.10	2.0
172	74.90	74.90	25.10	0.00	2.0
173	0.00	0.00	25.10	25.10	2.0
174	0.00	0.00	50.20	25.10	2.0
175	25.10	0.00	50.20	0.00	2.0
176	100.00	74.90	50.20	0.00	2.0
177	25.10	100.00	0.00	25.10	2.0
178	25.10	74.90	25.10	0.00	2.0
179	74.90	100.00	74.90	25.10	2.0
180	100.00	25.10	25.10	25.10	2.0
181	25.10	100.00	74.90	0.00	2.0



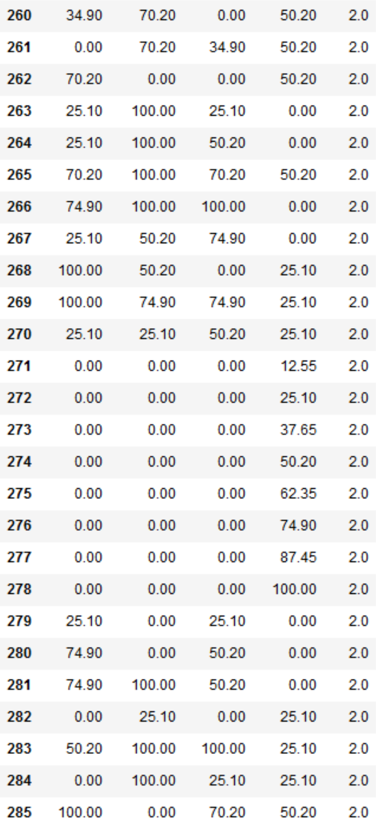
182	25.10	50.20	100.00	0.00	2.0
183	100.00	0.00	50.20	25.10	2.0
184	100.00	100.00	74.90	0.00	2.0
185	0.00	100.00	25.10	0.00	2.0
186	0.00	100.00	50.20	0.00	2.0
187	0.00	100.00	74.90	0.00	2.0
188	0.00	50.20	25.10	0.00	2.0
189	25.10	25.10	100.00	0.00	2.0
190	50.20	100.00	25.10	0.00	2.0
191	34.90	0.00	70.20	50.20	2.0
192	25.10	74.90	100.00	0.00	2.0
193	50.20	74.90	25.10	25.10	2.0
194	50.20	50.20	25.10	0.00	2.0
195	34.90	34.90	0.00	50.20	2.0
196	25.10	0.00	100.00	0.00	2.0
197	50.20	0.00	100.00	0.00	2.0
198	74.90	0.00	100.00	0.00	2.0
199	74.90	25.10	50.20	0.00	2.0
200	25.10	100.00	50.20	25.10	2.0
201	50.20	0.00	100.00	25.10	2.0
202	74.90	0.00	74.90	0.00	2.0
203	100.00	50.20	74.90	25.10	2.0
204	0.00	0.00	100.00	50.20	2.0
205	0.00	70.20	0.00	50.20	2.0
206	0.00	70.20	70.20	50.20	2.0
207	74.90	100.00	100.00	25.10	2.0



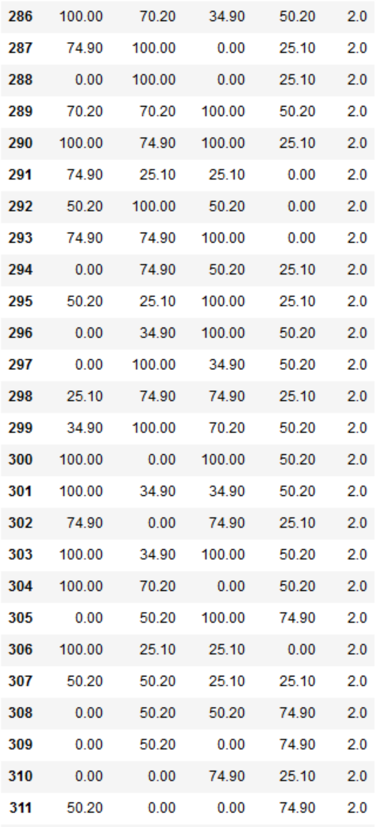
208	100.00	100.00	25.10	0.00	2.0
209	100.00	34.90	0.00	50.20	2.0
210	25.10	100.00	100.00	0.00	2.0
211	25.10	74.90	25.10	25.10	2.0
212	70.20	70.20	34.90	50.20	2.0
213	50.20	25.10	74.90	25.10	2.0
214	74.90	74.90	100.00	25.10	2.0
215	70.20	70.20	0.00	50.20	2.0
216	100.00	100.00	100.00	0.00	2.0
217	0.00	74.90	100.00	25.10	2.0
218	100.00	100.00	25.10	25.10	2.0
219	100.00	34.90	70.20	50.20	2.0
220	25.10	74.90	74.90	0.00	2.0
221	50.20	0.00	25.10	25.10	2.0
222	34.90	100.00	0.00	50.20	2.0
223	34.90	100.00	34.90	50.20	2.0
224	70.20	34.90	0.00	50.20	2.0
225	70.20	100.00	34.90	50.20	2.0
226	25.10	50.20	25.10	0.00	2.0
227	34.90	34.90	34.90	50.20	2.0
228	50.20	100.00	50.20	25.10	2.0
229	74.90	74.90	50.20	25.10	2.0
230	100.00	0.00	0.00	25.10	2.0
231	50.20	74.90	50.20	0.00	2.0
232	12.55	0.00	0.00	0.00	2.0
233	25.10	0.00	0.00	0.00	2.0



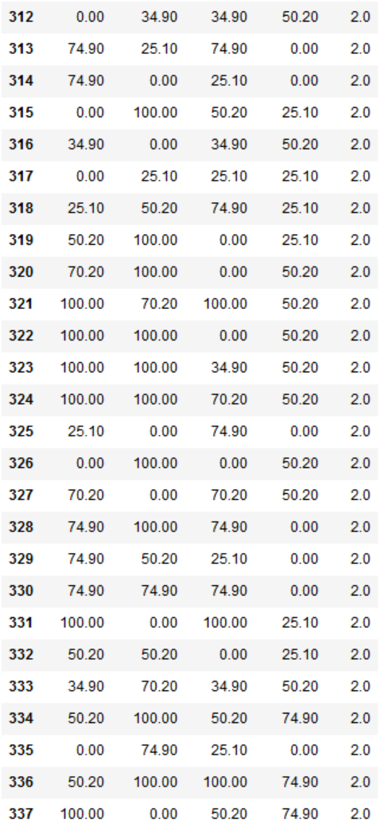
234	37.65	0.00	0.00	0.00	2.0
235	50.20	0.00	0.00	0.00	2.0
236	62.35	0.00	0.00	0.00	2.0
237	74.90	0.00	0.00	0.00	2.0
238	87.45	0.00	0.00	0.00	2.0
239	0.00	50.20	50.20	0.00	2.0
240	50.20	0.00	50.20	0.00	2.0
241	50.20	25.10	100.00	0.00	2.0
242	50.20	50.20	50.20	0.00	2.0
243	50.20	100.00	100.00	0.00	2.0
244	25.10	74.90	0.00	25.10	2.0
245	50.20	50.20	50.20	25.10	2.0
246	74.90	74.90	0.00	0.00	2.0
247	74.90	74.90	50.20	0.00	2.0
248	100.00	25.10	74.90	0.00	2.0
249	25.10	25.10	100.00	25.10	2.0
250	50.20	25.10	74.90	0.00	2.0
251	0.00	50.20	0.00	25.10	2.0
252	25.10	74.90	0.00	0.00	2.0
253	25.10	100.00	25.10	25.10	2.0
254	74.90	74.90	74.90	25.10	2.0
255	100.00	0.00	74.90	25.10	2.0
256	100.00	25.10	50.20	25.10	2.0
257	74.90	25.10	50.20	25.10	2.0
258	0.00	0.00	70.20	50.20	2.0
259	0.00	100.00	70.20	50.20	2.0



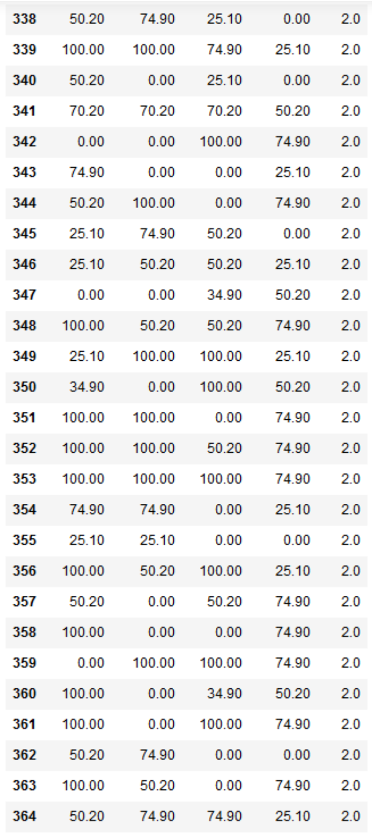
260	34.90	70.20	0.00	50.20	2.0
261	0.00	70.20	34.90	50.20	2.0
262	70.20	0.00	0.00	50.20	2.0
263	25.10	100.00	25.10	0.00	2.0
264	25.10	100.00	50.20	0.00	2.0
265	70.20	100.00	70.20	50.20	2.0
266	74.90	100.00	100.00	0.00	2.0
267	25.10	50.20	74.90	0.00	2.0
268	100.00	50.20	0.00	25.10	2.0
269	100.00	74.90	74.90	25.10	2.0
270	25.10	25.10	50.20	25.10	2.0
271	0.00	0.00	0.00	12.55	2.0
272	0.00	0.00	0.00	25.10	2.0
273	0.00	0.00	0.00	37.65	2.0
274	0.00	0.00	0.00	50.20	2.0
275	0.00	0.00	0.00	62.35	2.0
276	0.00	0.00	0.00	74.90	2.0
277	0.00	0.00	0.00	87.45	2.0
278	0.00	0.00	0.00	100.00	2.0
279	25.10	0.00	25.10	0.00	2.0
280	74.90	0.00	50.20	0.00	2.0
281	74.90	100.00	50.20	0.00	2.0
282	0.00	25.10	0.00	25.10	2.0
283	50.20	100.00	100.00	25.10	2.0
284	0.00	100.00	25.10	25.10	2.0
285	100.00	0.00	70.20	50.20	2.0



286	100.00	70.20	34.90	50.20	2.0
287	74.90	100.00	0.00	25.10	2.0
288	0.00	100.00	0.00	25.10	2.0
289	70.20	70.20	100.00	50.20	2.0
290	100.00	74.90	100.00	25.10	2.0
291	74.90	25.10	25.10	0.00	2.0
292	50.20	100.00	50.20	0.00	2.0
293	74.90	74.90	100.00	0.00	2.0
294	0.00	74.90	50.20	25.10	2.0
295	50.20	25.10	100.00	25.10	2.0
296	0.00	34.90	100.00	50.20	2.0
297	0.00	100.00	34.90	50.20	2.0
298	25.10	74.90	74.90	25.10	2.0
299	34.90	100.00	70.20	50.20	2.0
300	100.00	0.00	100.00	50.20	2.0
301	100.00	34.90	34.90	50.20	2.0
302	74.90	0.00	74.90	25.10	2.0
303	100.00	34.90	100.00	50.20	2.0
304	100.00	70.20	0.00	50.20	2.0
305	0.00	50.20	100.00	74.90	2.0
306	100.00	25.10	25.10	0.00	2.0
307	50.20	50.20	25.10	25.10	2.0
308	0.00	50.20	50.20	74.90	2.0
309	0.00	50.20	0.00	74.90	2.0
310	0.00	0.00	74.90	25.10	2.0
311	50.20	0.00	0.00	74.90	2.0



312	0.00	34.90	34.90	50.20	2.0
313	74.90	25.10	74.90	0.00	2.0
314	74.90	0.00	25.10	0.00	2.0
315	0.00	100.00	50.20	25.10	2.0
316	34.90	0.00	34.90	50.20	2.0
317	0.00	25.10	25.10	25.10	2.0
318	25.10	50.20	74.90	25.10	2.0
319	50.20	100.00	0.00	25.10	2.0
320	70.20	100.00	0.00	50.20	2.0
321	100.00	70.20	100.00	50.20	2.0
322	100.00	100.00	0.00	50.20	2.0
323	100.00	100.00	34.90	50.20	2.0
324	100.00	100.00	70.20	50.20	2.0
325	25.10	0.00	74.90	0.00	2.0
326	0.00	100.00	0.00	50.20	2.0
327	70.20	0.00	70.20	50.20	2.0
328	74.90	100.00	74.90	0.00	2.0
329	74.90	50.20	25.10	0.00	2.0
330	74.90	74.90	74.90	0.00	2.0
331	100.00	0.00	100.00	25.10	2.0
332	50.20	50.20	0.00	25.10	2.0
333	34.90	70.20	34.90	50.20	2.0
334	50.20	100.00	50.20	74.90	2.0
335	0.00	74.90	25.10	0.00	2.0
336	50.20	100.00	100.00	74.90	2.0
337	100.00	0.00	50.20	74.90	2.0



338	50.20	74.90	25.10	0.00	2.0
339	100.00	100.00	74.90	25.10	2.0
340	50.20	0.00	25.10	0.00	2.0
341	70.20	70.20	70.20	50.20	2.0
342	0.00	0.00	100.00	74.90	2.0
343	74.90	0.00	0.00	25.10	2.0
344	50.20	100.00	0.00	74.90	2.0
345	25.10	74.90	50.20	0.00	2.0
346	25.10	50.20	50.20	25.10	2.0
347	0.00	0.00	34.90	50.20	2.0
348	100.00	50.20	50.20	74.90	2.0
349	25.10	100.00	100.00	25.10	2.0
350	34.90	0.00	100.00	50.20	2.0
351	100.00	100.00	0.00	74.90	2.0
352	100.00	100.00	50.20	74.90	2.0
353	100.00	100.00	100.00	74.90	2.0
354	74.90	74.90	0.00	25.10	2.0
355	25.10	25.10	0.00	0.00	2.0
356	100.00	50.20	100.00	25.10	2.0
357	50.20	0.00	50.20	74.90	2.0
358	100.00	0.00	0.00	74.90	2.0
359	0.00	100.00	100.00	74.90	2.0
360	100.00	0.00	34.90	50.20	2.0
361	100.00	0.00	100.00	74.90	2.0
362	50.20	74.90	0.00	0.00	2.0
363	100.00	50.20	0.00	74.90	2.0
364	50.20	74.90	74.90	25.10	2.0

Appendix H

Media Used in Training Data

The following media are used in the project (and ultimately in the training dataset).

Serial Number	**Supplier**	**Medium Name**	**Weight**	**Coating**	**Country**
1	G-print	Arctic Paper	115 gsm	Matt	EU
2	G-print	Arctic Paper	300 gsm	Matt	EU
3	Verso	Blazer Satin Text	148 gsm	Silk	US
4	UPM	Digi Finesse Premium Silk	115 gsm	Silk	EU
5	UPM	Digi Finesse Premium Silk	300 gsm	Silk	EU
6	Sappi	Magno Plus Gloss	115 gsm	Gloss	EU
7	Sappi	Magno Satin	115 gsm	Silk	EU
8	Sappi	Magno Satin	300 gsm	Silk	EU
9	Sappi	Flo Digital Dull Text	148 gsm	Silk	US
10	Sappi	Flo Digital Dull Cover	270 gsm	Silk	US
11	Sappi	Magno Gloss	115 gsm	Gloss	EU
12	Sappi	Magno Gloss	300 gsm	Gloss	EU
13	UPM	Digi Gold Gloss	130 gsm	Gloss	EU
14	Sappi	Flo Digital Gloss Text	118 gsm	Gloss	US
15	Sappi	Magno Matt	115 gsm	Matt	EU
16	Sappi	Magno Matt	300 gsm	Matt	EU
17	Sappi	Magno Plus Silk	115 gsm	Silk	EU
18	Sappi	Magno Volume	135 gsm	Matt	EU
19	Sappi	Magno Volume	250 gsm	Matt	EU
20	Sappi	Mccoy Gloss Cover	325 gsm	Gloss	US
21	Sappi	Mccoy Silk Cover	270 gsm	Silk	US
22	Verso	Sterling Premium Gloss Cover	271 gsm	Gloss	US
23	Verso	Sterling Premium Silk Cover	271 gsm	Silk	US
24	Canon	Symbol Card 2SC	300 gsm	Silk	US

Table H.1: Media included in the data set