University of Vaasa

Faculty of Technology

Communication and system engineering

Kongadzem Eve-Mary Leikeki

MACHINE LEARNING APPLICATION: ORGANS-ON-A-CHIP IN PARALLEL

Master's thesis for the degree of Master of Science in Technology that has been submitted for inspection, Vaasa 26 April, 2018

Supervisor	Prof. Mohammed Elmusrati
Instructor	Prof. Nureddin Ashammakhi Shaima Abdelmageed

LIST OF CONTENTS

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
ART	Adaptive Resonance Theory
SOM	Self-organizing map
CHAID	Chi-squared Automatic Interaction Detection
MARS	Multivariate Adaptive Regression Splines
ID3	Iterative Dichotomiser 3
SVR	Support Vector Regression

UNIVERSITY OF VAASA
Faculty of technology
Author: Kongadzem Eve-Mary Leikeki
Topic of the Thesis: Machine learning application: Organs –in-a- chip in parallel
Supervisor: Prof. Mohammed Elmusrati
Instructor: Prof. Nureddin Ashammakhi
Shaima Abdelmageed
Degree: Master of Science in Technology
Major Subject: Communication and Systems Engineering
Year of Entering the University: 2015
Year of Completing the Thesis: 2018

ABSTRACT

Cancer is one of the most common deadly diseases in the world. This is due to the fact that no good cure is available. Many medications developed for this disease are studied in vitro and then tested in animals (such as rats, pigs, sheep, and in some cases primates). Most of this testing prove successful but may fail to do so in clinical trials (in human body). The failure of the drug testing during the clinical trial is a result of the dissimilarity between animals and human. By developing a human tissue or organ-on-a-chip, the structure, physiology and functions of a human organ can be represented on a chip. When many of these human organs are integrated in parallel on-a-chip-devices, a high throughput screening system for drug discovery is obtained. However, this will involve the production of large amount of data which will require artificial intelligence, neural networks or machine learning data analytic methods to analyze and derive reliable conclusions. Produced data consist of the numeric values for the parameters necessary for testing drug efficacy (such as the solubility of drug and iron levels in the blood stream) that are measured from the integrated chip. Machine learning is the most affordable and reliable data analytic method which learns from a training data in order to predict or classify an output based on a similar test data. The training data in this case are the exact numerical values of the parameters necessary for determining the drug efficacy with their expected cell reactions which were defined after the drug was developed while the test data are the measured numerical values of the parameters from the integrated chip whose cells reactions to the drug need to be analyzed. This thesis therefore aim at the outlining the present method used for testing the efficacy of developed drugs, the reasons why these drugs fail in clinical trial and the use of organs–on-a-chip-in parallel as an improved method for the drug development process. Data is generated with respect to the values of parameters used in determining the efficacy of a drug such as Cisplatin (a drug for treating cancer) and a support vector machine learning algorithm is trained to test these parameters in order to classify and predict the reactions of cells. This will give rise to the claim that if multiple human organs are developed on-a-chip and integrated on-a-chip-devices, the integrated devices will act like a human body and when contaminated with a disease, a developed drug can be scan through the integrated chip to identify the reactions of the cells to the drug thereby determining the drug efficacy. This will reduce the time spent in testing drugs as well as the risk of mortality during clinical trials or even after drug approval and use.

Keywords: Data, Organs, Parameter, Chip, Model

1 INTRODUCTION

Cancer is one of the leading deadly diseases in the world. This disease has kept every one worried since there is no curative therapy for most of cancers. Surprisingly, developed drugs which were studied in vitro and tested successfully in pigs failed during the clinical trial to treat humans (Nicole 2016). This is due to the dissimilarities between human and pigs. Approximately, $2.6 billion is used to develop and market a single drug which normally takes an average of ten years (Aaron 2014). The whole process is very lengthy and expensive which could lead to failure at the clinical trials. Since the activities, mechanics or physiological response of a human organ such as kidney, lungs and guts can be stimulated by a multichannel 3D or 2D microfluidic system (organ-on-a-chip), it is expected that when many human organs are developed on chips and integrated on-a-chip-devices, the integrated chip will eventually function as a normal human body and testing drugs with this method will be very effective. This will involve the production of large amount of data since the values for the parameters necessary for determining the drug efficacy need to be measured from the cells in the integrated chip. These parameters are analyzed using machine learning technique in order to classify or predict the cells reactions to the drug in the integrated chip thereby determining drug efficacy.

Machine learning techniques use different types of algorithms to learn from a training data in order to classify or predict an output based on a similar data sample in case of supervised learning or where there is no training data and the algorithm groups the data in terms of similarities or the closeness in distance as with the case of unsupervised learning. Data sample from health institute have few classes which can be grouped as positive response to treatment, negative response to treatment, no effect on treatment or inaccurate treatment. This data must be analyzed with certainty. Thus, it requires a supervised machine learning technique whose algorithms such as support vector machine can learn from a training data in order to classify or predicts the reactions to treatment or body cells (that have an influence in treating a particular disease) to the drug, there by determining the efficacy of the drug.

The current project outlines the present procedure for drug development, challenges involved in using this process and how the process can be improved by developing multiple human organs on different chips and integrating them on a chip device. A simulation is also provided in this

project in order to learn and predict the cell reactions (these reactions are classified as dead cells, unaffected cells, negative response or positive response) to the drug in the chip based on the values of the measured cell parameters necessary for determining drug efficacy.

1.1 Motivation

I lost my uncle due to cancer so I kept asking myself why they have not been any cure for this disease until now and how it can be prevented and treated. A brief discussion with professors Nureddin Ashammakhi and Mohammed Elmusrati outlined the fact that most of these incurable diseases proved successful in animal test but failed in clinical trials due to the dissimilarity between humans and animals. They further said that if multi organ-on-a-chip platform is developed and integrated it may eventually lead to better drug development. After several researches to this claim, I found out that this solution of testing drugs with organ-on-a-chip integrated in parallel could be possible since the toxicity from the drug compound can be screened with a multichannel microfluidic cell (organ-on-chip) to stimulate the activities, mechanics or physiological response of human tissues/organs. By developing disease model-on-a-chip of the disease to be treated, a drug can be inserted and screened through the integrated chip in order to measure the values of the parameters necessary for drug efficacy from the cells in the integrated chip. This will involve the production of large amount of data which are analyzed using machine learning techniques in order to determine the efficacy of the drug. Thus, the reason for the claim that multiple organ–on–a–chip devices in parallel can be used to determine the efficacy of drugs at low cost, minimum risk and within a short period of time

1.2 Objectives

The main aim of this thesis is to forecast that after integrating many developed organ–on-a-chip in an integrated chip device and using them to test for drug efficacy, time, cost and risks involved in testing the developed drugs will be reduced. When the drug is applied into the integrated chip, the values for the parameters necessary for determining the drug efficacy are measured from the cells of each developed organ-on-a-chip on the integrated chip in order to identify the cells reactions to the drugs at specific body functions such as excretory function or waste removal (Kidney-on-a-chip), digestive function (gut-on-a-chip), respiratory function (lungs-on-a-chip). This will therefore give an analysis of the cells reactions to the drug by the respective organs-on-a-chip in the integrated chip. This analysis is done with the use of machine learning technique where the values of the measured parameters from the cells of each developed organ-on-a-chip in the integrated chip are tested with a trained model in order to classify the cell reactions to the drug in the integrated chip device thereby determining drug efficacy.

1.3 Methods

Large amount of data is extracted from the integrated chip consisting of many developed human organs-on-a-chip that are integrated on-a-chip devices. This data is analyzed and classified using different machine learning algorithms which learns from a sample labeled data in order to predict the cell reactions to the drugs in the integrated chip. The type of machine learning algorithm used for predicting the cells reactions depends on the values of measured parameters (such as the values of solubility, blood partitioning into each organ, iron level in the blood stream) from the cells of each developed organ-on–a-chip in the integrated chip. When there are few classes of observed reactions of cells to the drug (as in the case of this project where the cells reactions are classified in to four classes of dead cell, unaffected cells, negative response to the drug or positive response to the drug) and each class having a reasonable amount of measured

parameters, then a support vector machine learning algorithm can be used to predict the cells reactions to the drugs. If this condition does not hold, then another algorithm is used based on the values obtained from the measured parameters. In this project, a sample medical data containing the measured values of different cells parameters necessary for determining the drug efficacy is used in training a support vector machine learning algorithm thereby building and training a model. The trained model is used to predict the reactions of the cells to the drug in another similar data sample. This analysis is done through 'R project for statistical computing' where a training data and test data are provided and a computational analysis is done between the two data sets using support vector machine learning algorithm. Thus, when different human organs are developed on a chip and integrated on-a-chip-devices, this integrated chip can be corrupted with a disease and by inserting a drug in the integrated chip; the cells parameters necessary for determining the drug efficacy can be screened and measured. A trained model with a similar labeled data is then used to predict the cells reactions of these measured parameters to the drugs in the integrated chip. The interpretations of these cell reactions will determine the efficacy of the drug.

1.4 Thesis Structure

This Thesis is divided into five parts which include the following: firstly, the introduction which gives a brief description about the thesis and the reason for this particular research. Secondly, the literature review is presented. In this part, the present method used in testing for drug efficacy is outlined. Furthermore, the third part gives an overview of machine learning as being able to learn and make predictions from data without being programmed. Part four provides an outlined of Support Vector Machine (SVM) as a machine learning algorithm that will be used in training and testing the data set in this project. Also, a simulation to train a model to predict the reaction of cells to a drug will be done using the SVM on "R project for statistical computing" in this chapter. Lastly a conclusion and further recommendation will be provided in chapter five.

2 LITERATURE REVIEW

After the development of drugs or biomaterials, they have to be tested first by using cell culture to screen for their toxicity. Once proved nontoxic, the next step will include testing in animal (general models by administration or implantation, for example in the subcutis or special models for defined ultimate application) to assess *in vivo* safety biocompatibility in the body, feasibility of the application and success for the intended purpose. Once these preclinical studies prove successful, the next testing level is the clinical trials.

There are three different phases of drug testing on human which include; phase I, clinical trials in which small numbers of patients are used and the objective is to assess safety in humans. Phase II is the clinical trials where larger number of patients are recruited and the objective is to assess the drug efficacy. In phase III, the largest numbers of patients are included to further study the safety and the efficacy of the drug with different populations and dosages. Once completed with these phases, the application for approval by regulatory body is submitted. After approval obtained for the product, the drug or biomaterial can be marketed. (Fang & Richard 2017)

2.1 Toxicity Test with Cell Culture

In vitro testing generally consists of a biological system (cultured permanent or primary cells that are derived from type-culture collection and explants respectively), the cell-material contact (which could be direct or indirect) and the biological end points (for example vital staining, cytosolic enzyme release and cell growth). Drug toxicity can be screened by estimating the basal functions of the cells or by testing for specialized cell functions. Specialized cell functions are best assessed in testing the toxicity of chemicals that have already been established and in circumstances where *in vitro* investigation of target tissues has been successfully clarified by

specific cell cultures while cell basic function are processes common to all the cell types and support certain cell functions. (Schmaiz 1994.)

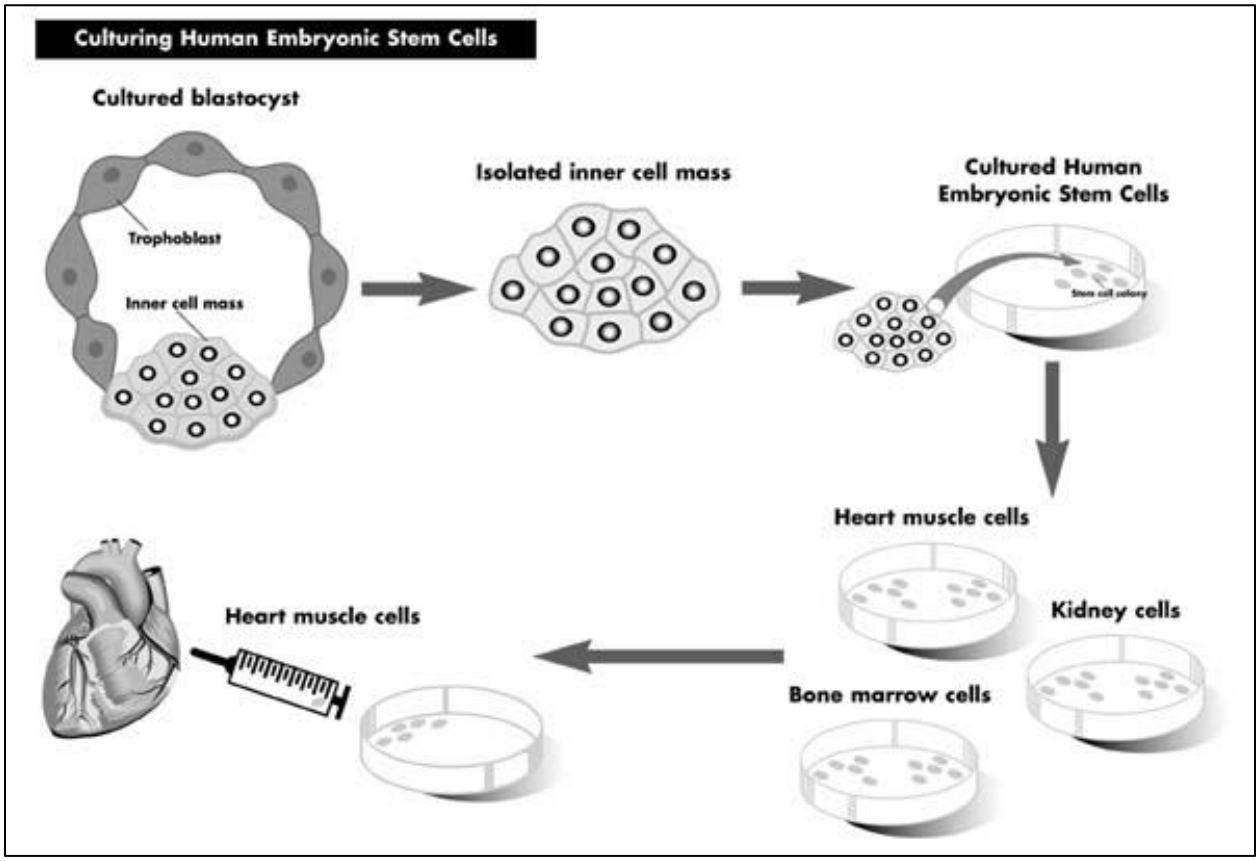


Figure 1. The culture of human stem cell (Sean 1998).

Stem cells are important cells used to remove toxicity from drugs. Once they are extracted, from humans, they are kept in a culture medium which then differentiate into endoderm, ectoderm and mesoderm germ lines. These germ lines differentiate into the heart muscle, kidney and bone marrow cells which are directed to specific cells by creating unlimited amount of virtual cells that can serve as research for heart, Parkinson's and many cancer treatments. (Sean 1998)

2.2 Technologies Involved in Cell Culture

Cells grown under controlled environment are good candidates for testing drugs. Previously, monolayer cell cultures were used for cell screening and drug discovery. The advancement in 2D cell models enabled their usage in drug discovering and prediction response of drugs to different agents. 2D cell models have limitations due to the loss of tissue specific architecture, mechanical and biochemical signs thereby making them unable to predict the drug response of complicated disease such as cancer. The recent implementation of 3D cell cultures alongside with the use of stem and primary cells enables easier prediction of efficacy and possible toxicity in humans. The cells in 3D culture are grown into 3D aggregates using a scaffold and are able to replicate the extracellular matrix which is able to provide the physical structure and to facilitate cellular communication. Examples of 3D culture technologies include spheroids, organoids, engineered tissues, and most recently organs-on-chips where grown 3D tissues are continuously perfused through micro channels in the structure. (Fang & Richard 2017)

2.2.1 Spheroids

The availability of oxygen gradient, nutrients, metabolites and soluble signals in spheroids cultures enables the compensation for deficiencies thereby creating many heterogeneous cells. Also, there is a good interaction between cell-cell and cell-extracellular matrix (a collection of extracellular molecules from cells which are capable of providing structural and biochemical support to the neighboring cells) using spheroid cultures. These two characteristics of interaction and deficiency compensation have led to the use of tumor spheroid monocultures alongside with immune cells to study cancer disease development and for drug screening. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

Four approaches used in spheroid cultures include the following; the first approach involves the use of low-adhesion plates to enable the self-assembly of cells into spheroidal clusters. The low attachment surface coating as well as the shapes of the plates bottoms (round, tapered or v-shaped) minimizes the cell adherence and eases the positioning of spheroids within themselves respectively. This approach therefore, helps in obtaining a high-throughput screening since the spheroids are formed, propagated and assayed in the same plate. The second approach uses a hanging dropping plates in order to increase the formation of many cellular spheroids. Cells dispensed on the top of the hanging drop plates are separated into discrete media droplet at the bottom of the plate which eventually becomes spheroids. The limitation of this approach is the transfer of hanging drop plate from one plate to another for evaluation. The third approach is the use of bioreactors (spinner flask or microgravity bioreactor) to initiate the self-splitting of cells into spheroids. This approach enables large-scale production of spheroids, but it has a limitation of non-uniform size of spheroids. The last approach is the cultivation of spheroid through the use of micro-nano-patterned surfaces that controls cell adhesion and migration. Different scales of scaffolds are imprinted on the flat substrate in order to select the pattern and the adhesive properties for the cell types. The main disadvantage to this approach is the formation of bubbles during the cultivation process and the damages of micro patterned surfaces during pipetting. (Fang & Richard 2017.)

Some challenges involved in the use of spheroid culture include the following: firstly, the problem of developing and maintaining spheroids of uniform size. Secondly, there is a limitation of forming spheroids from cells that have small seeds. Furthermore, during the process of co-culture, the difficulty involved in controlling specific ratios of different types of cells in a spheroid is faced. Lastly, spheroid cultures are unreliable, difficult, non-standardized and have high-throughput which makes them unsuitable for drug screening. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

2.2.2 Organoids

Organoids are artificially grown masses of cells or tissues. The artificial growing of tissue is known as tissue organoids which refer to the processing of connecting tissues of different organs that are free of cell cultivation. Epithelial cells are example of tissue organoids with the ability of independently organizing into tissue–like structures. The growth of stem cells is known as stem cell organoids. They are generated from embryonic stem cells (ECSs) or from induced pluripotent stem cells (iPSCs). Examples of existing organoids and their characteristics include the following; firstly, the thyroid organoid used as a functional organoid is cultured by combining the embryoid bodies in hanging drops. Secondly, gastric organoids which originate from adults stem cell as well as from the gastric glands are cultured by Matrigel embedding and used in adult stem cells and in developing all stomach epithelial pathologies except gastric cancer. Furthermore, kidney organoid connects the tubes or pipes that are surrounded by the renal interstitial space located within the tissue or organ and endothelial cells. Kidney organoids are sub cultured after differentiation and dissociation in an air to liquid media. In addition, thymus organoids are used in the transplantation of thymus epithelial cells and are cultured by inducing Forkhead box protein N1 (FOXN1). Lastly, retinal organoids which are cultured from V- shaped SFEBq (serum free floating culture of embryonic body aggregates that have quick reaggregation) plates, embedded with Matrigel by the second day, and by the twelfth day they are transferred to a Petri dish and used in epithelial as well as in determining the retinal pathologies. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

The limitations involved in using organoids in testing for drug toxicity include the following; firstly, there is the absence of vasculature which is important in the transportation of nutrient and waste. Secondly, some organoids do not have certain cell types found in normal living body. Furthermore, some organoids do not have the capability of replicating at later stages of organ development. For example, there is no outer segment in the retinal organoids and the photoreceptors are not light sensitive since they do not get mature. Lastly, the primary control of the cell fate and its behavior occurs externally which might be different from that found in animals. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

2.2.3 Scaffolds and Hydrogels

Scaffolds are 3D synthetic structures made of materials that can be porous, permeable and have mechanical characteristics in order to resemble the microenvironment of native tissues. Two types of scaffolds include biological and polymeric scaffolds. Biological scaffolds are derived from extracellular matrix such as Matrigel and Collagen which permit the attachment and reorganization of cells into 3D structures. The microenvironment provided by Matrigel has molecules, hormones and a soluble growth factor that enables cells to interact *in vivo* environment but the limitation involved with the use of scaffolds material is that of manufacturing variations and complexity in composition which make it difficult in determining the signal that is promoting the cell function. Polymeric scaffolds on the other hand comprise synthetic hydrogels or other materials that are biocompatible in order to create the physical support for 3D cell cultures. Poly (ethylene glycol), poly(vinyl alcohol) and poly(2-hydroxy ethyl methacrylate) are hydrogels used for 3D cultures which can be hydrolyzed or biodegraded enzymatically by including lactic acid in the polymer network backbone. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

The use of polymeric scaffolds in 3D cell culture has greater advantages when compared with biological scaffolds. They include the following: firstly, polymeric scaffolds do not need to be reproduced since they are simply processed and manufactured as compared to biological scaffolds that have a poor reproduction of biological extracellular matrix thereby leading to inconsistency between cultures. Secondly, mechanical and chemical properties are optimized in 3D cell cultures with polymeric scaffolds since polymer permits the tuning of mechanical and biochemical properties. Furthermore, polymeric scaffolds contains a lot of water that permits transportation of oxygen, nutrients, waste and soluble factors which are essential to cell functions. Lastly, polymeric scaffolds are able to regulate cell behavior. However, polymeric scaffolds have their limitations such as follows; firstly, the lack of endogenous factor (internal forces that occurs within cells). Secondly, there is in availability of oxygen within the polymeric scaffolds. Furthermore, a synthetic cellular microenvironment is heterogeneities and lastly, the

distribution of soluble growth factor within the matrix is not uniform. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

2.2.4 Three-Dimensional Bioprinting

This is the process of using additive manufactures to print cells, biocompatible material and other components into complex 3D living constructs. Biological materials, biochemicals and living cells are positioned layer after layer using the following approaches. Firstly, the biomimicry approach which aims at replicating cellular and extracellular components of a tissue or organ. Secondly, the approach of self-assembly which produces biological micro architecture and functional tissues by the method of differentiating the cells into specialized tissues and organs during growth. The last approach is fabricating and assembling building blocks (for example kidney nephron) in to larger design by method of rational design, self-assembly or by combining the two methods. (Rasheena, Jessica, Broglie, Audrey &Yang 2014.)

3D bioprinting techniques are generally preferred to scaffold bioprinting because of the following reasons: firstly, 3D bioprinting implies that tissue construct development in 3D bioprinting is not affected by the rapid increase of cells within the scaffolding materials, whereas with the scaffold bioprinting techniques, the development of tissues depends on the increase of cells within the scaffolding materials like hydrogel. Secondly, 3D bioprinting have high initial cell density prior to the inclusion of biomaterials which facilitates the deposition of extracellular matrix thereby resulting to a good cell-to-cell interaction, tissue generation within the biomimicry and phenotype cell preservation. Lastly, 3D bioprinting can use spheroids in order to fabricate thin tubular tissues such as blood vessels and nerve grafts. Despite the above-mentioned benefits of using 3D bioprinting, some limitations are still faced such as follows: firstly, 3D bioprinting consumes a lot of energy. Secondly, particles are emitted into the air when 3D bioprinting techniques are used and lastly, the cell environment is difficult to maintain thereby leading to the death of many cells. (Rivero James & Ibrahim)

2.2.5 Organ–On-A–Chip Platforms

The activities, mechanics or physiological response of a human organ such as kidney, lungs and gut can be stimulated by a multichannel 3D or 2D microfluidic cell structure known as organ-on-chip. These cell structures are designed with the capability of rebuilding the structure, microenvironment and functions of the human organs. The chips have well-defined structures, patterns or scaffolds made of microfabrication techniques (for example lithography, photolithography and contact printing) that have a good precision. Using microfluidics of high space-time precision, the position, shape, functions, physical and chemical microenvironments of the cells to be cultured can be determined. The cells on chips are able to respond to stimuli and to perform realistic physiological functions. Advantages involved in using organ-on-a-chip *in vitro* physiological models include the following: firstly, organ-on-a-chip devices generate fluid shear stress which provides stable nutrients as well as discharging timely waste. Secondly, the dynamic mechanical stress generated with organ-on-a-chip by using flexible porous membrane and cultured cells can mimic certain physiological functions such as lung breathing, intestinal peristalsis and heart contractions. Furthermore, the streamline flow of fluid produces concentrated gradients which are beneficial in physiological processes such as cell migration, differentiation, immune response and cancer metastasis. Lastly, the control of pattern cells is realized with organ-on-a-chip which is beneficial in constructing *in vitro* physiological model of a complex geometric structure. (Sun, Chen, Luo. Min, Zhang & Wang 2016.)

When air enters the lungs, it fills the alveoli then oxygen is transferred through the lung cells membrane into the bloodstream. The lung cell membrane is a thin, flexible and permeable membrane that is made up of a three interfaced layer of lung cells, a pervious extracellular matrix and a capillary blood vessel cells. This interface between the lung and blood is able to recognize inhaled bacteria and toxins. It can activate immune response as well. Lung-on-a-chip micro devices consist of two layers of cells which include lung's air (gas channel) sacs lining and blood vessels (liquid channel). These two channels are separated by a porous elastic membrane (made of polydimethylsiloxane (PDMS)). Air is passed to the lung lining while blood is initiated

by the culture medium that flows in the capillary channel. Breathing is initiated by the cyclic mechanical stretching. (Elizabeth Dougherty 2010.)

The kidney is an essential excretory organ responsible for maintaining the osmotic pressure and the internal body environment. Kidney-on-a-chip device can be used to replace the lost function of the kidney. Nephron is an important functional unit of the kidney whose responsibility is to remove metabolites and waste through the production of urine and retain moisture alongside the re-absorption of other nutrients. Using a double layered chip separated by an elastic porous membrane the kidney-on-a-chip can be designed. The function of the upper layer is similar to that of a urinary lumen and it has a fluid shear stress that affects cell function and morphology while the lower channel contains media and has an intermediate space for the mechanical system of the chip. Kidney-on-a-chip can be used to perform the following: firstly, to improve the dose of drugs in kidney diseases. Secondly, the use of kidney-on-a-chip can aid in understanding the increase in urea blood and other nitrogenous waste. Furthermore, kidney-on-a-chip can help in drug testing and development. Lastly, it can be used to test and develop drugs for kidney diseases. (Jonathan &Joseph 2013.)

Gut-on-a-chip is concerned mainly with the digestive and absorptive functions. Since most drugs are taken into the human body orally which passes through the small intestine into the blood circulation. Gut-on-a-chip can be used to study the absorption of drugs through the intestinal cells which is very important in drug screening. The small intestines villi built on the chip helps in studying the function of the intestine such as absorption of nutrients. (Sun, Chen, Luo. Min, Zhang & Wang 2016)

The modes of motion involved in the intestinal tract include segmental contraction, intestinal peristalsis which plays a significant role in nutrient digestion and absorption. Gut-on-a-chip designed is capable of imitating the structure of the gut and modes of the intestinal tract as shown in the figure below. (Richard 2012)

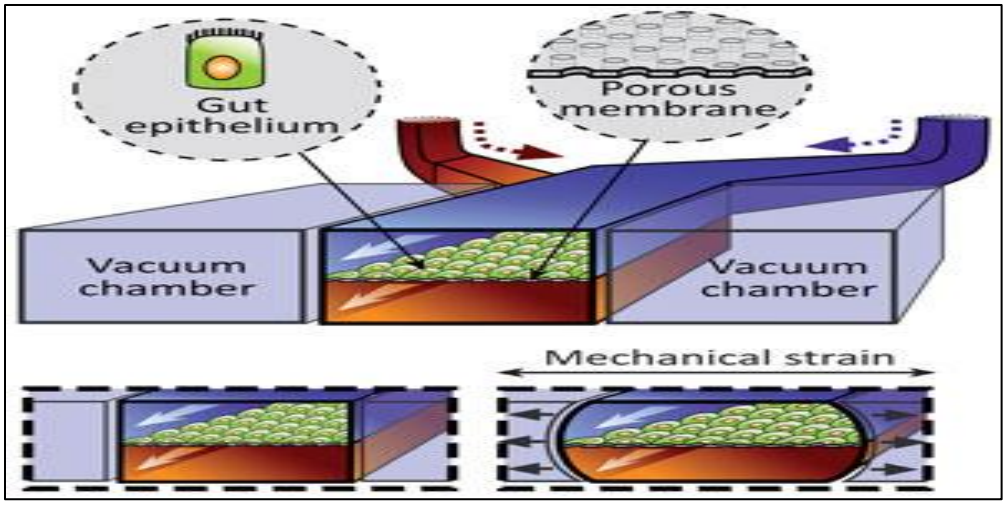


Figure2. Gut-on-a-chip (Richard 2012).

Figure2 above shows gut-on-a-chip made of two microfluidic channels that are separated by a porous flexible membrane. Extracellular matrix is used to coat this porous membrane and the human intestinal epithelial cells are lined on the membrane. The passage of the fluid into the tube at a low rate enables the creation of a natural gut's environment. Air is removed from the vacuum chambers in order to force fluid thereby exerting a mechanical strain on the tube at a constant interval which imitates the functions of intestinal peristalsis (absorption of nutrients). The above characteristics of the gut-on-a-chip lead to the development of a columnar epithelium which is similar to the intestinal villi structure. By growing normal intestinal microbiome on the surface of the epithelium, a biomimetic model could be designed and used for both absorption and toxicity studies, drug tests and development of new intestinal disease models. (Richard 2012.)

The above-mentioned cell culture techniques are useful in restoring the function and micro environmental features of human tissues and cells. Thereby, making them a good choice for drug toxicity studies and possibly replace animal testing in future.

2.3 Drug Testing in Animals

When drug toxicity is rolled out after cell culture testing, the drugs are tested in animals. This testing entails a hypothesis that can be accepted or rejected. If accepted, then it serves as a basis for further hypothesis and if rejected then a new hypothesis will be developed. However, animals are from different species and cannot always have reactions similar to those that may occur in human (Silvio &Giuliano2016.)

Testing procedure is done by developing disease models in animals. The animals in this model could be used to find out different medication aspect such as using some of the animals to test the toxicity of the chemical compound of the drug, checking if the drug can actually cure the disease and checking the consequence (side effect) of using the drugs after a long period. (Prasad 2016.)

Although drugs to be used for bacterial, fungal or viral infections are effectively tested using this drug development procedure, some diseases such as stroke, amyotrophic lateral sclerosis (the degeneration of the system motor neurons central nervous) and Alzheimer's disease do not still have a good translation from animals to humans due to differences between species. Apart from this in compatibility in the translation of drugs from animal to man, there exist certain limitations of animals for testing drug such as follows: firstly, it is expensive, time-consuming and unreliable. Furthermore, the differences in toxin metabolism, chemical and drug absorption of certain animal species have resulted to erroneous or inadequate information regarding the application of animal data to human disease and drug response. Lastly, there can be contamination of results in the lab. Since animals undergo stress which influences heart function, pulse, muscular activities and blood pressure which can modify the values of the variable thereby producing wrong drug analysis. (Neavs 2018)

As a result of the ineffectiveness of testing drugs in animals, new methods are being explored in order to improve patient's treatment as well as reducing the number of deaths caused by the drug

failure. One of such research yielded to the idea of organ-on-a-chip in parallel where multiple organ-on-a-chip devices are developed in order to produce similar structure, functions and physiology like that of the human body. Thus, when the parameters necessary for determining drug efficacy are measured from the integrated chip and analyzed, cellular response to drugs can be known thereby determining the drug efficacy or predicting the causes of drug failure in human body.

3 ORGANS –ON-A-CHIP IN PARALLEL

In order to develop a system that can predict the success of a developed drug in human body, multi organ-on-a-chip devices are developed. The integrated chip in turn resembles human body. This is an important setup. For example, the kidney is sensitive to the toxicity effects of drugs as a result of waste products excretion and compounds re-absorption while the liver changes the concentration and bioavailability of drugs taken orally. Due to the fact that many drugs change their toxicity level after the metabolism in the liver, liver cells and cells of other organs are required to be investigated together in order to obtain the metabolic effect of the respective organs. For example in order to study the toxicity of a cancer drug, liver and kidney cells were incubated for a day, then the drug to be tested was added on the chip and the system was perfused for three days. (Sun, Chen, Luo, Min, Hong-Yang, Wang)

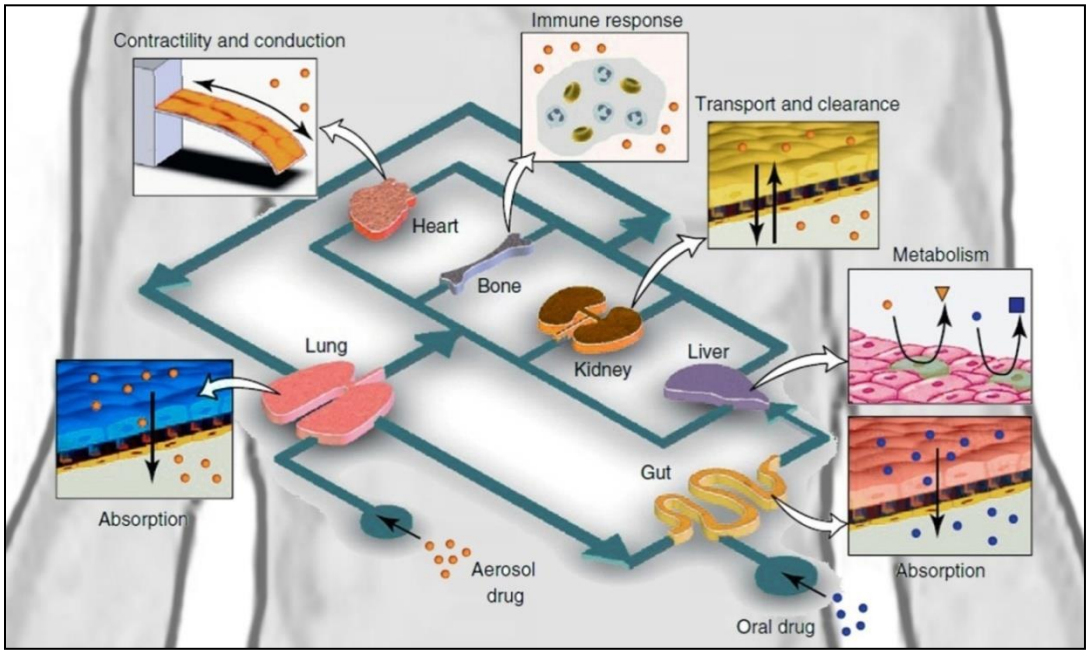


Figure3. Multi organ-on-a-chip in which various organ-on-a-chip devices were integrated in a microfluidic compartment (Marjorie, Julie &Guilhem).

Figure 3 shows six human organs integrated in a microfluidic compartment. Each organ in the microfluidic compartment has a specific characteristic that represents the functions of that organ

in the human body. For example, oral drugs are absorbed at the gut and the metabolism of this drug is found in the liver as well as kidney used for clearance and transport. This figure represents a sample layout in which organs-on-a-chip in parallel will be integrated on-a-chip devices but in this case, the developed organ-on-a-chip will be positioned in place of the respective organs (this means instead of placing the lungs, liver and gut in the microfluidic compartment, the developed lungs-on-a-chip, liver-on-a-chip and gut-on-a-chip will be placed in an integrated chip device respectively). This will enable correct analysis of the drug at each organ without interactions from neighboring organs thereby overcoming the challenge of the interconnected systems in the chip not being similar to the interactions of organs/tissues of the human body as in the case when multiple human organs are arranged in the microfluidic compartment without developing them on chips.

By including a disease in an integrated chip, a developed drug can be applied on the integrated chip and a sensor used to virtualize the performance of the drug in the chip. This will result in recording the measurements of the parameters necessary for determining the drug efficacy from the integrated chip. Analyzing these parameters with machine learning techniques, the cells reactions to the drug in the integrated chip can be determined thereby determining the drug efficacy. This chapter will outline the concept, types and applications of machine learning.

Machine learning evolved from pattern recognition and computational learning theory where algorithms were constructed in order to learn and predict data. The interaction between pattern recognition and computational learning aspect of machine learning is important because as new data are introduced into the model, they are recognized and independently adapted. Therefore machine learning is a data analytical method that speedy the rate of model building by training the machines to learn and adapt through experience without being programmed. The use of computer science in machine learning is beneficial for two reasons. Firstly, training the model to sort the data appropriately, to process it and finally save the data. Secondly, computer science enables the effective representation of inference data sample as well as ensuring the efficiency of the algorithmic solution used in learning the model. By analyzing complex and big data, machine learning can provide precise models (predictive or descriptive models) that can identify different types of drugs or avoid the risk of using certain drugs for medical treatment. (Thomas 2017.)

Two main tasks of machine learning are supervised and unsupervised machine learning. These tasks are classified based on the availability of feedback to the learning system. Tasks that require feedback to the learning system are computed with supervised machine learning while those that do not require to provide feedback to the learning system are classified with unsupervised machine learning. Some tasks based on the desired output of the machine-learned system include the estimation of numerical value, prediction of a group of data, finding natural groups of data, querying similar objects and finding the frequency of objects. (Ajitesh 2015.)

3.1 Unsupervised Learning

This type of learning aims at finding the regularities in the input without any supervision. This is structured in such a way that the occurrence of some patterns is more than others. An example of an unsupervised learning is clustering where natural groups of data are found and labeled. Image compression is an application of clustering. In this application, the image pixels represented as RGB values are the input variables. Similar colors are grouped using a clustering program. These groups correspond to the frequently occurring colors in the image, taking the average of these groups implies quantizing the image while the analysis of repeated image pattern such as texture and objects enables a simple and higher-level description of the image. Grouping similar documents for example news report related documents (sports, fashion, and arts) is known as document clustering. In this case, N words are predefined and every document is an N-dimensional binary vector having an element ’i’ that is assigned to 1 if it is sorted in the document. The usage of words such as “of”, “and” are omitted from the documents since there are not informative as well as the removal of “-s” and “-ing” suffixes in order to avoid duplication. (Ethem 2014.)

Unsupervised learning algorithms use the hidden features in the data in order to group the data. Data grouping enables the features in the raw data that were hidden to be known.

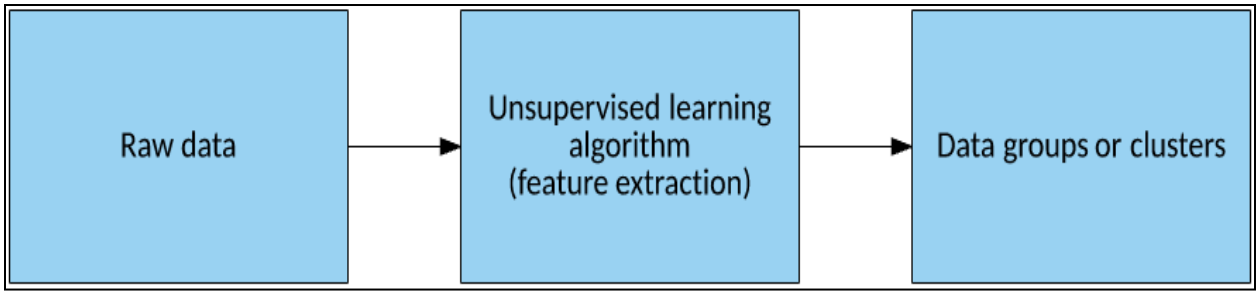


Figure4. Unsupervised Learning (Tim 2017).

Figure4 shows the process involved in sorting data using unsupervised learning. The data is first fed and an algorithm is used to extract the unique features of the data which classifies them. Data with the same features are grouped under same clusters or groups. (Tim 2017.)

3.1.1 Unsupervised Machine Learning Algorithms

Adaptive Resonance Theory (ART), Self-organizing map (SOM) and K-means are examples of algorithms used for clustering. Using a weight vector, an ART1 algorithm maps an input vector into a neuron at a recognition field. The main advantage of ART is the ability of incorporating the vigilance parameter to determine the closeness of the vector characteristic to the mapped cluster in order for it to be a member. SOM is used to convert many dimensional feature vectors into a low dimension. In SOM, the data is compressed using vector quantization and its map is made up of a neural network. Individual neurons have a unique weight vector that matches with the dimension of the input vector. K-means algorithm organizes the feature vectors (any set of observation) into clusters based on the distance measurement of k center of the cluster. The closest centre is the cluster in which the feature vector is a member. Initialization, Update and assignment are the three steps for a training algorithm of a k-mean as shown in the diagram below. (Tim 2014.)

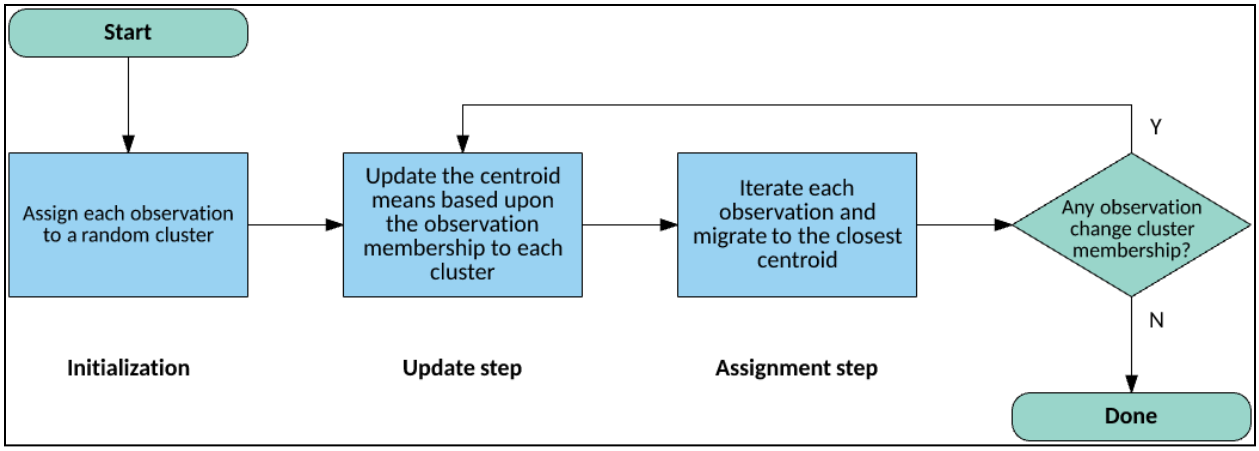


Figure5. K-means clustering algorithm (Jones 2017).

As shown in figure5, when observations or data are inputted, they are first assigned randomly to any cluster. Then the centre mean of the cluster is updated based on the observation membership to each cluster in the update step. The assignment step ensures that the observation or data is in the correct cluster by calculating the Euclidean distance of the data or observation's characteristic vector with each k cluster centre. The observation or data is then placed in the closest cluster. If there is no change in the observation or data's cluster after the assignment step then the grouping is completed and if some observations change their clusters then the update step as well as the assignment step is repeated. (Jones 2017.)

3.1.2 Applications of Unsupervised Machine Learning

Applications of unsupervised learning include the following; firstly, segmentation in the market is possible with unsupervised machine learning where customers are grouped according to their buying activities thereby acquiring information for the discounts organized by the company. Secondly, unsupervised learning can be employed in the bank to sort out fraud. Furthermore, the grouping of documents based on their contents as well as image segmentation can be done with the use of unsupervised machine learning. In addition, the data of the earth science can be grouped using unsupervised learning thereby obtaining the climatic condition. Lastly, audio

recordings captured through the microphone and placed in specific region of interest uses unsupervised machine learning to analyze the records in order to estimate the biodiversity (for example species of birds, animals) in that region. (Sunil 2018.)

3.2 Supervised Learning

This type of machine learning uses an algorithm that learns the mapping function from the input to the output. It is known as supervised learning because the process of making an algorithm to learn from its training dataset can be regarded as a teacher supervising the learning process. Therefore, in supervised machine learning, the correct data (output) is known while the algorithm makes several predictions on the testing data and it is corrected by the teacher (training data). The learning process could be for a single class (positive and negative examples) or multiple classes that can be classified under many classes. (Ethem 2014.)

3.2.1 Learning from a Class

In order to learn a class, the descriptions for all positive and negative examples of the class are studied. The features of the class that distinguish between positive and negative examples are referred to as inputs. For example, taking the toxicity of a cancer drug as a class and solubility level of the drug as inputs, drugs that contains the required levels of toxicity and solubility from the given set of drugs will be considered as good drugs for cancer treatment while any other drug will be regarded as bad samples without necessarily checking the type of treatment which they could be suitable for. Each drug can then be represented based on the level of toxicity and solubility. (Ethem 2014.)

Let the drug toxicity be represented as x1 and x2 represented as drug solubility. Labeling the treatment as r, the following equations can be obtain

(3.1) $$X = \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

(3.2) $$r = \begin{cases} 1 & if\ x\ \ can\ treat\ cancer \\ 0 & if\ \ x\ \ cannot\ treat\ cancer \end{cases}$$

Considering N training set of drugs and each drug represented as an ordered pair (X, r) as follows

(3.3) $X = \{X^t,\ r^t\}$ where t refer to the different types of drugs from 1 to N

The assumption that a particular drug can treat cancer is from the fact that the toxicity level and the solubility level are in an accepted range as proposed when the drug was developed. This is represented mathematically by the following equation;

(3.4) $(t1 \leq \text{toxicity} \leq \text{t2})\text{AND}\ (s1 \leq \text{solubility} \leq \text{s2})$

Equation 3.4 provides a better hypothesis class H which the learning algorithm tries to locate a specific hypothesis h within it (h ε H) in order to approximate the drug.

Therefore, the aim is to find h ε H that is similar as possible to the drug class.

(3.5) $$h(x) = \begin{cases} 1 & the\ classification\ of\ x\ is\ positive \\ 0 & the\ classificationo\ f\ x\ \ is\ negative \end{cases}$$

During training, if the predictions of h does not march the required value given in X, an empirical error will occur. This error is given by

(3.6) $E(h|x) = \sum_{t=1}^{N} (h(X^t) \neq r^t)$

(3.7) $E\left(\{h_i\}_{i=1}^{k}|x\right) = \sum_{t=1}^{N} \sum_{i=1}^{k} 1\,(h_i(X^t) \neq r^t)$ where i = 1,…,k, are different set of classes

Equation 3.6 is the error obtain when a single class is used while 3.7 is that obtain when k classes are used. When one of the $h_i\ (x)$ is 1, then a class can be chosen. But if they are more than two specific hypotheses with the value of one or if none of the hypothesis have the value one then a class cannot be chosen. (Ethem 2014)

Each hypothesis h from H is defined by one quadruple $(t_1^h, t_2^h, s_1^h, s_2^h)$ and the values of these parameters most be found in the training data with the inclusion of the positive and negative examples. A problem may arise by generalizing the hypothesis since the future data whose characteristics are closed to the boundary between positive and negative examples may result to different predictions from sample hypothesis. This problem can be solved using the most specific hypothesis that includes all the positive examples and no negative one. (Ethem 2014)

The presence of noise in the data makes it more difficult for the class to be learned thereby resulting to errors in the prediction. This noise is as a result of the following; firstly, the impreciseness of the input recordings which results to a data shift. Secondly, the errors in the data points labeling may result to fault representation of the data (for instance, positive situation might be represented as negative and negative instances as positive). Lastly, hidden or latent unobservable attributes that were not considered before labeling are modeled as random components which are considered as noise. (Ethem 2014.)

Complex models with larger parameter as well as simple models with less training errors can be used to reduce this noise in the data. Complex models gives a perfect fit to the data and are able to illuminate the error in it. This is done in the expense of larger parameter, where as simple models use less parameter and permits certain errors in the model. Simple models are often preferred to complex models because points within the model can easily be identified as well as reliably predicting future instances. Also, the learning of a simple model result to the extraction of a raw data from the training set. (Ethem 2014.)

3.2.2 Classification of Supervised Learning

The aim of supervised machine learning problems is to construct an excellent model that can predict the dependent value of an attribute from a list of variable attributes. These problems are grouped into regression and classification. Regression problems have real or continuous output variable (for example weight or salary). Regression models are classified as linear and non-linear regression which could be simple or multiple (1 or 2 features respectively). Linear regression is the simplest of this model with the capability of fitting the data within the best hyper-plane that goes through the points. Classification problems on the other hand use categories (such as red, yellow or sick) as an output variable. Classification models are able to predict outputs from observed values by using the training set and class labels of the classified attributes. Examples of classification models include logistic regression, decision tree and Naïve Bayes. (Bansal 2017.)

3.2.3 Techniques Used in Supervised Machine Learning Algorithm

Linear regression, logistic regression, decision tree, SVM and Naïve Bayes are some algorithms used in training the machine in order for it to make specific decisions. The machine trains itself continuously using trial and error method thereby being able to learn from past experience and to provide accurate results. These algorithms are discussed in detailed below. (Sunil 2017.)

Linear regression is a supervised machine learning algorithm that uses continuous variables to estimates numerical values (weight, number of calls, salary). The relationship established between the dependent and the independent variable is through a regression line (a best fit line). Simple and multiple linear regressions are the two types of linear regression. Simple regression has one independent variable while multiple regressions have more than one independent variable and when finding their best fit line, a polynomial or curvilinear regression is used. (Sunil 2017.)

(3.8) y = a*x + b

Equation 3.8 represent the regression line where

y = dependent variable

a = slope

x = independent variable

b = intercept

By taking the difference in distance between the data points and regression and minimizing their sum of squared, the coefficients ‘a’ and ‘b’ are found.

Logistic regression uses independent set of variables to estimate discrete values. Logistic regression therefore predicts the probability of an event occurring by fitting data into a logic function. The difference between linear regression and logistic regression is that the modeled of the output value in logistic regression is a binary value (0 or 1) while that in a linear regression is based on numeric value. (Jason 2016.)

(3.9) $$y = e^{(b_0+b_1*x)} / \left(1 + e^{(b_0+b_1*x)}\right)$$

Equation 3.9 is an example of a logistic regression equation where y is the predicted output, b_0 is the intercept term while b_1 is the coefficient for the independent value (x). b_0 and b_1 are real values that are learned from the training data.(Jason 2016.)

(3.10) $$P(X) = P(Y = 1|X)$$

Equation 3.10 is a mathematical text explaining the model probability that an input (X) belongs to a class (Y = 1) by default.

Considering the above logistic regression, it can be proven that the output is a linear regression as shown below

(3.11) $$p(x) = e^{(b_0+b_1*x)} / \left(1 + e^{(b_0+b_1*x)}\right)$$

Dividing the both sides by 1- $p(x)$, equation 3.11 becomes

(3.12) $$\frac{p(x)}{1-p(x)} = \left[e^{(b_0+b_1*x)} / \left(1 + e^{(b_0+b_1*x)}\right)\right] / 1- \left[e^{(b_0+b_1*x)} / \left(1 + e^{(b_0+b_1*x)}\right)\right]$$

Further simplification shows that

(3.13) $$\frac{p(x)}{1-p(x)} = e^{(b_0+b_1*x)}$$

Taking natural logarithm of both sides yields the following equation

(3.14) $$\ln\left(\frac{p(x)}{1-p(x)}\right) = (b_0 + b_1 * x)$$

Equation 3.14 proves that the regression function is a linear method but the logistic function is used to transform the predictions. The left ratio of 3.14 is called the odds of the default class and it is calculated as the ratio of the event probability divided by the probability that the event do not occur. Since this ratio is transformed by the log function, the left hand side of equation 3.14 is called log-odds. (Jason 2016.)

Hence the equation

(3.15) $$\ln(odds) = (b_0 + b_1 * x)$$

It can therefore be concluded from equation 3.15 that the model is a linear combination of the inputs whose outputs are the log-odds of the default class.

A decision tree is an active model capable of predicting the values of a specific target correctly based on the input variable created. Decision trees are built by checking the base of the model, iterating all attributes in the data, normalizing the information gain from splitting through all the attributes and creating decision node that splits on the nodes with the highest gain. Iterative Dichotomiser 3 (ID3), C4.5 algorithm, Chi-squared Automatic Interaction Detection (CHAID) and Multivariate Adaptive Regression Splines (MARS) are different algorithms developed for

the analysis of the decision tree. Each of these algorithms is suitable for different specific types of data. ID3 calculates the entropy of all the attributes in the dataset then make a decision based on the node with the highest gain. It is best used in smaller trees because the training data fits very closely to a limited set of data. C4.5 algorithm calculates the threshold point for splitting the attributes. It is therefore an improved algorithm over ID3 because it has the ability to work with missing attribute values. In this case a question mark is indicated on the missing values and their gain and entropy are simply skipped. CHAID algorithm uses large sample sizes of data which can be split in many ways. It is mostly applicable in marketing to select and predict how a group of consumer response to certain variable which can affect other variables. MARS on the other hand is used for numerical data. It is an extension of linear models that interact between linear and nonlinear models. (Jason 2015.)

Decision tree can be used on both discrete and continuous dependent variables. An independent variable which is regarded as the most significant attribute is used to split different attributes into two or more homogenous set as shown in the example below.

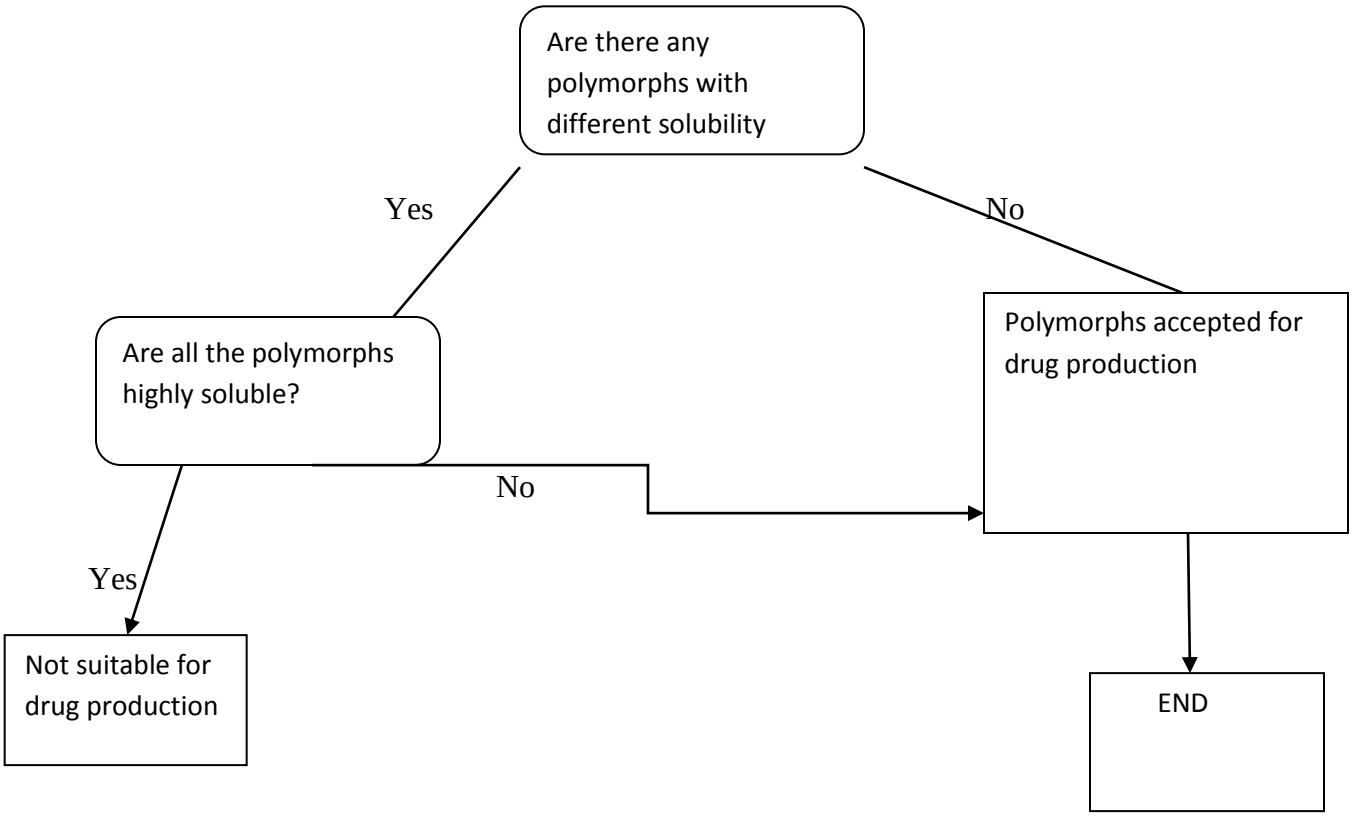


Figure 6. Decision Tree for polymorphs solubility (Lawrence 2002).

Figure 6 above shows a sample decision tree to test for polymorphs solubility. Firstly, the polymorphs solubility is being tested and if there are no polymorphs with different solubility level then the polymorphs is suitable for drug production and end. Otherwise if there were polymorphs with different solubility then a test is performed whether all the polymorphs are highly soluble or not. In case some polymorphs are not highly soluble, they will be considered as being effective in producing drugs. While those that were highly soluble will be regarded as not being suitable for drug production and discarded. (Lawrence 2002.)

Naïve Bayes machine learning algorithm is based on Bayes rule (determines the previous knowledge of an event condition and described the probability of that event based on it). It works on a principle that two features in a class are unrelated. Though this algorithm is simple, it can be used for sophisticated model and large data. (Sunil 2017.)

(3.16) $$P(h|d) = \big(P(d|h) * P(h)\big)/P(d)$$

Where

$P(h|d)$ Called the posterior priority, is the probability of obtaining a hypothesis h when a data d is given.

$P(d|h)$ Is the probability of having a data d when the hypothesis h is true

$P(h)$ Called the prior probability, is the probability of the hypothesis h being true

$P(d)$ Is the probability of the given data

3.16 is the equation for the Bayes theorem where there is an interest in obtaining the posterior probability. When these posterior probabilities of different hypotheses are calculated, the hypothesis with the highest probability often referred to as Maximum a Posteriori (MAP) is selected. As seen in this equation, the representation for Naïve Bayes is with the conditional and class probabilities. The conditional probabilities represent the tentative probability of individual input given the value of each class while the class probabilities represents the individual class probabilities in the training data. (Jason 2016.)

Data training with Naïve Baye machine learning is fast because coefficients are not used thereby reducing the time required in fitting them. Predictions of new data for a Naïve Bayes model are also possible using Bayes theorem.(Jason 2016.)

Support Vector Machine (SVM) is another machine learning algorithm that is used in classification, regression and detection of data. It creates a boundary that separates two groups of data based on the provided set of training data. Advantages of SVM include the following; firstly, it is beneficiary when used in high dimensional spaces. Secondly, it has an effective memory which facilitates the use of the data training point's subset in the decision function. Lastly, there are many varieties of kernel functions that can be employed in the decision function. Despite the above benefits of using SVM, there still have the challenge of not providing probability estimates directly. Five-fold-cross-validation is used to calculate these probabilities. Since medical data are mostly non linear (need to be mapped in a higher

dimensional space) and do not need probability estimates, SVM have been highly employed in medical image recognition for training and predicting disease based on the images provided to train the model. (Scikit-Learn)

Another application of SVM in medicine is seen in this project where it will be used to classify and predict the reactions of cells to a drug inserted in a chip. More details about SVM machine learning algorithm will be outline in chapter 4

3.2.4 Applications of Supervised Machine Learning

Some applications of supervised machine learning include its use in systems biology where it can express gene using microarray data, text categorization in order to detect spam, face detection in order to discover customer and in medicine to predict the probability of a patient having a heart disease by analyzing the spectrum of his/her ECG. (Gideon, Isabelle &Fishel 2008.)

Microarray is a technology that is used to monitor large number of genes outline in parallel. It is made up of thousands of DNA sequences which bind together when complemented and does not bind when non-complemented (For example a 10 nucleotides consisting of TACCGAACTG will hybridize with the sequence ATGGCTTGAC where the ‘A’ nucleotides complements the ‘T’ and the ‘C’ complement the ‘G’). This usually involve a wide generation of data and with the help of machine learning, the complexity provided by this bulky data is overcome. The procedure involved in microarray gene expression is shown in the figure below. (Patrick & Lucila 2004.)

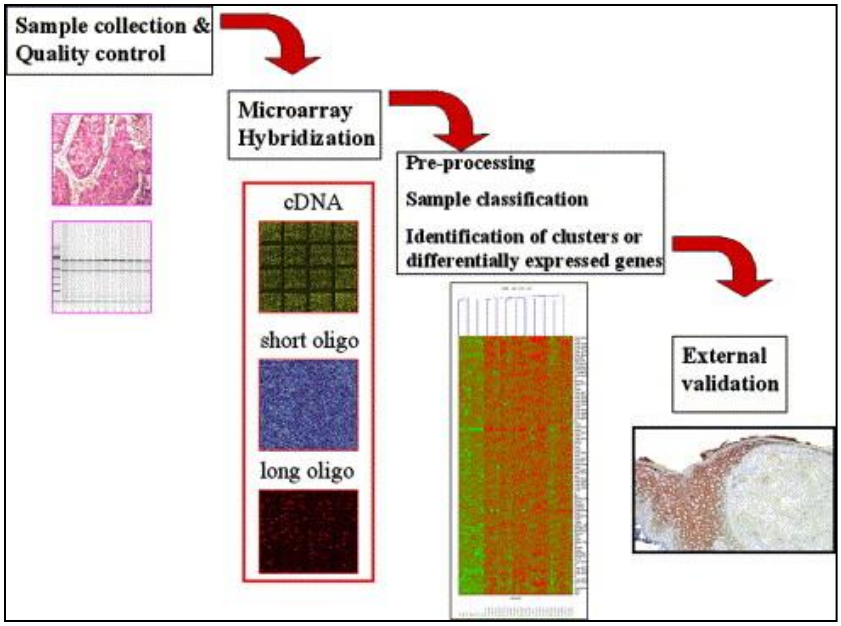


Figure 7.Gene expression procedure (Patrick & Lucila 2004).

Tissue or cells samples are collected and the mRNA is extracted from the samples and used cDNA is produced from it as shown in figure 7. The cDNA binds to the sequences that does not mobilized on the microarray while the sequences that did not bind will be washed away. The next stage is the pre-processing stage which involves the classification of samples, clusters identification and expressed gene differentiation. The use of machine learning algorithm in this stage is of great benefit since it enables the huge and complex data to be easily identified. The algorithm can be train to group genes according to specific characteristics. Based on these characteristics, the machine learning trained algorithm will be able to group future genes thereby leading to a perfect gene expression of various kinds of data. (Patrick & Lucila 2004.)

Supervised machine learning can categorize texts in to different form. For example news can be categorized in the form of sports news, political news as well as science news. This manner of arranging news organizes a magazine thereby making it easier for the reader to understand the message that is being passed across. The training and prediction phases are the two modes which are used to classify the text. In the training phase, a human being gives the correct classification of the text which will be train with one of the machine learning algorithm, while the prediction

phase is when the actual classification is done based on the input text and the learned algorithm. (Gideon, Isabelle &Fishel 2008.)

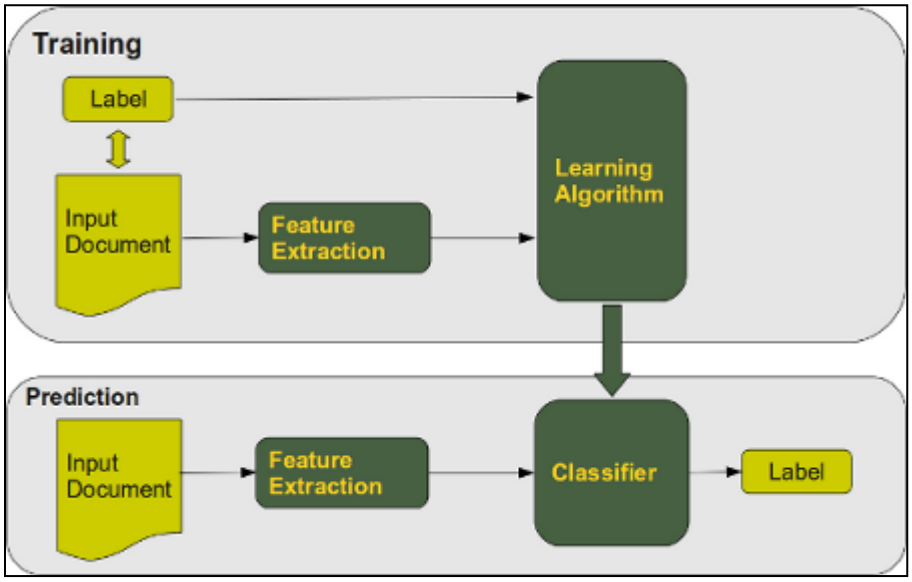


Figure 8. Using machine learning algorithm to categorize documents. (Bern, Bodenseo& Denise 2011).

As seen in figure 8, in the training phase, the features are extracted from the label input data (a classified document) and used to teach the machine learning algorithm. The prediction phase use a more general document without labels whose features are extracted and classified according to the experienced of the learned algorithm and then labeled based on the information from the algorithm. (Bern, Bodenseo& Denise 2011.)

Another important application of supervised machine learning is in face detection. Facial features unique to a particular person are extracted and stored in a compact feature vectors. A machine learning algorithm such as SVM is then used to train the model. SVM generate and distinguish the model based on several data classes. A single class in facial recognition represents a unique face and the SVM has a function of separating the multiple features of this class from those of other unique face (class) by finding the hyper plane that separates the largest

fraction of the points within the same class while ensuring that the distance from this plane to the class is large. (Arun & Peter 2010.)

The importance of supervised learning is also seen in signature recognition where it recognizes signatures using structural similarities. The machine has to learn two set of signatures, one that contains genuine and forged signatures of many individuals (general learning) and another that contain multiple samples for a single person genuine signature (special learning). Features are extracted from these two types of learning and their similarities are computed. In general learning, these features are extracted from the genuine-genuine and the genuine-forgery pair of sets and the distance between the pair is computed by measuring their similarities. The special learning on the other hand uses multiple genuine signatures whose variations are learned in order to effectively determine a genuine signature that is within the range of the variation. The distance between the features of the special learning samples are distributed by comparing the similarities of the known samples. (Harish, Sargur & Matthew 2006.)

3.3 Semi Supervised Learning

Semi supervised learning is a class of supervised learning that uses unlabeled data for its training. The data in this technique is divided into two parts, one part with a labeled training and another without labels. The main advantage of using semi supervised learning is due to the fact that unlabeled data is cheap and labeled data might be hard to get because of the need for experts in providing them. In order to generalize a finite training set, a semi supervised smoothness assumption which applies to both regression and classification most be made. In case of classification, the unlabeled data helps to find the boundary of each cluster while the label points are used to assign a class for an individual cluster. The cluster assumption therefore suggests that if points are of the same clusters then they will surely be of the same class. Manifold assumption is good for classification as well as regression. Manifold of higher dimensions are placed above those of low dimensions in order for the learning algorithm to operate in the space of the

corresponding dimension thus minimizing the problem of the rapid growth in volume as a dimension is increased. (Olivier, Bernhard & Alexander 2006.)

3.3.1 Characteristics of Problem Solvable by Semi Supervised Learning

Semi-supervised learning is not suitable for solving all scenarios. Specific characteristic are required to be present on the problem before they can be considered solvable by semi supervised learning. They include the following: firstly, the size of the unlabeled data should be greater than the labeled data. In case the labeled data is greater, then the problem can be solved with supervised learning. Secondly, the input and output should be closely symmetry. This is because the deduction of the classified unlabeled data is based on the closeness with the labeled data points. Therefore, if two data points are of the same clusters then their outputs are expected to be closed as well. Lastly, the problem should be easy with simple labels and few dimensions. Problems with complex label and those that use many data sets of many dimensions are very challenging for semi-supervised learning. (Rodriguez 2017.)

3.3.2 Applications of Semi Supervised Learning

The application of semi supervised learning includes speech recognition, webpage classification and protein sequences. Speech recording usually involve huge amount of speech which are strenuous for an individual to listen and type its transcript, thereby making it difficult to have labeled data for speech analysis. Thus, the application of semi supervised learning techniques can improve the traditional models for speech analysis. Web pages can easily be processed automatically but their classification into different headings or titles will require a lot of manual labor for human being to sort them. This can be avoided by using semi supervised machine learning techniques. The use of semi supervised learning in protein sequences just like in other

applications reduces the time and energy spent by human in resolving a three-dimensional structure or in determining the function of each single protein. (Rodriguez 2017.)

4 SUPPORT VECTOR MACHINE

Support Vector Machine is a type of supervised machine learning algorithm that locates a decision function in order to create a boundary that can separate two groups of training data. The selection of the decision function is based on distance between the test data and the closest training data. When data is not separated linearly by a decision function, a kernel transformation function is used to map the data into different dimensional space so that they can be linearly separated using the standard decision function techniques of support vector machine. (Brandon 2011.)

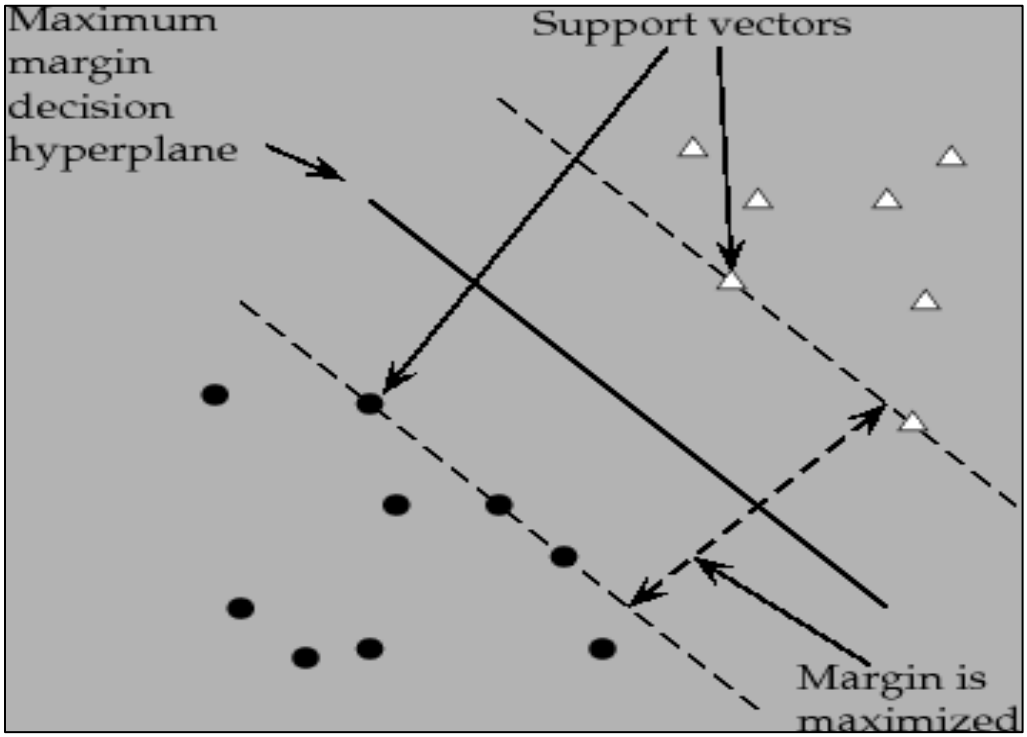


Figure9. A linear separable case of support vector machine. (Christopher, Prabhakar & Hinrich 2008.)

Figure 9 above shows an example of a support vector machine whose case is linearly separable. The bold line is the decision function while the 5 points at the dotted line are the support vectors. Using a large margin (the distance from the decision function to the support vector), the model

capacity is reduced and the points can be clearly classified without any uncertainty. (Christopher, Prabhakar & Hinrich 2008.)

Support Vector Machine supports both classification and regression tasks with continuous or categorical variables. The classification task separates different classes of labels by constructing hyperplanes in a multidimensional space, whereas regression task involve the determination of a certain function that have all points within a certain distance from those point. The regression tasks therefore utilize the support vector regression (SVR). SVM construct hyperplanes by repeating the training algorithm in order to minimize the error function. Depending on the error function form, SVM models can be classified into classification and regression where each model has a type1 and type 2. (Statsoft.)

In classification SVM type1, the error function is set to be minimized in the training data. Mathematically, classification SVM type 1 can be represented as

(4.1) $\frac{1}{2}w^T w + C\sum_{i=1}^{N} A_i$

Subjected to $y_{i(w^T \emptyset(x_i)+b)\geq 1-} A_i and A_i \geq 0, i = 1, \ldots, N$

Where

C = capacity constant

w = coefficient vector

A_i= parameters in charge of non separable data

N = number of training cases

i = Label cases

b = a constant

x_i= independent variables

y$\epsilon \pm 1$ is the class label

$\emptyset =$ The kernel used to transform data from the input to the higher dimensional feature space.

Errors in this type of classification can be minimized by reducing the capacity constant. This will intend reduce over fitting. (Statsoft.)

Classification SVM type 2 also known as nu-SVM aims at minimizing the error function in the model. It is similar to classification SVM type 1 but its capacity constant ranges from 0 to 1 whereas that of type1 is from 0 to infinity.

(4.2) $\frac{1}{2}w^T w - v\rho + \frac{1}{N}\sum_{i=1}^{N} A_i$

Subjected to $y_{i(w^T \emptyset(x_i)+b)\geq \rho -} A_i and A_i \geq 0, i = 1, \ldots, N and\ \rho \geq 0$

Where

V$\epsilon(0, 1]$. It controls the support vectors and the training errors.

ρ is an independent term.

One-classification is another type of SVM classification that is used for novelty detection. It separates all the data points in high dimensional space and maximizes the distance from the hyperplane to the origin. A binary function captures sections in the input space that returns +1 when the training data points are captured and -1 elsewhere. (Roemer 2013.)

4.3 $min_{w,A_i,\rho}\frac{1}{2}||w||^2 + \frac{1}{vn}\sum_{i=1}^{n} A_i - \rho$

Subjected to

$(w.\emptyset(x_i)) \geq \rho - A_i$ for all i = 1,…n

$A_i \geq 0$ for all i = 1,…n

Equation 4.3 is a quadratic function for the one-class SVM. The parameter v is the support vector at the lower bound of the training data. It therefore set an upper bound for the training data that are out of class. Support vector data description (SVDD) is used to model data with spherical boundary in order to find the smallest hypersphere that contains all the data samples. This hypersphere has a center ‘a’ and a radius R> 0 (distance from the center to any support vector) whose volume R^2 is minimized in order to reduce the incorporation of the training data outside of the studied class. (Roemer 2013.)

With regression SVM, the model is trained on a sample data set in order to find a function that can predict new cases that have not yet been presented in the model. Support vector regression is a method of support vector machine used in solving regression problems. SVR aims at minimizing error and maximizing the margin. By mapping the non linear learned machine into high dimensional kernel induced feature space, a non-linear function can be learned by the SVR. (Statsoft 2018.)

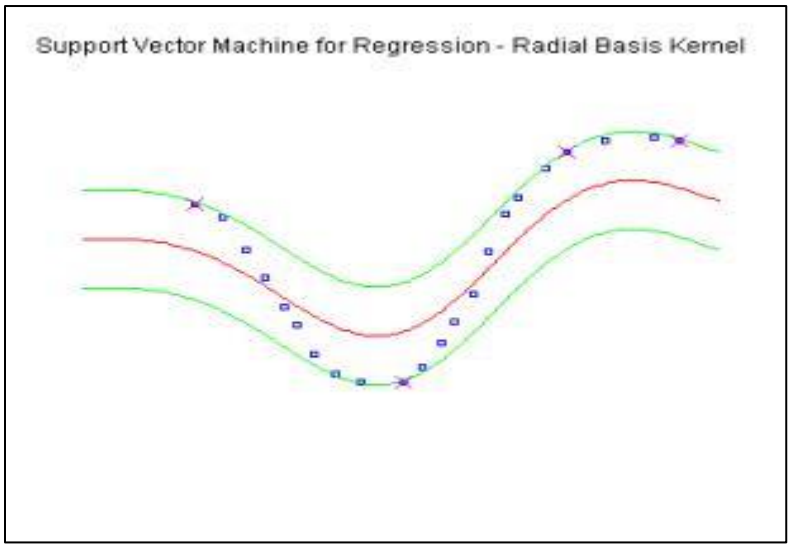


Figure10. Support Vector Machine Performance during a regression problem (Tripod).

Figure 10 shows the performance of a support vector machine during a regression problem. The green lines represent the epsilon boundary which defines the loss function that ignores errors, while the blue points are the trained data. In general, the parameters cost (C), insensitive loss (ε) and the kernel parameters determines the performance of an SVM model trained to solve a regression problem. 'C' creates a balance between the complexity of the model and the tolerance level in which the deviations is larger than the insensitive loss function. The parameter $'\varepsilon'$ determines the wideness of the insensitive zone that is used to fit the training data. The bigger the insensitive loss parameter, the smaller the support vector used thereby reducing the complexity of the model. A suitable kernel parameter will map the non linear data into a higher dimensional

feature space where a linear model can be constructed. Therefore the choice of kernel parameter will affect the trained model. (Tripod.)

4.1 Support Vector Machine in Cancer Drug Discovery

Assuming that a particular drug is applied to an integrated chip (a chip consisting of many developed human organs-on-a-chip integrated in parallel) that is infected with a cancer disease, the connected sensor will screen the reaction of drugs in the integrated chip to measure the values for the parameters necessary for determining the efficacy of drugs. Based on these measurements, different voltages are created which are digitalized with the software in order for them to be available for the computer. The computer uses machine learning algorithms in order to classify and predict the cellular response to the drug based on these measured parameters.

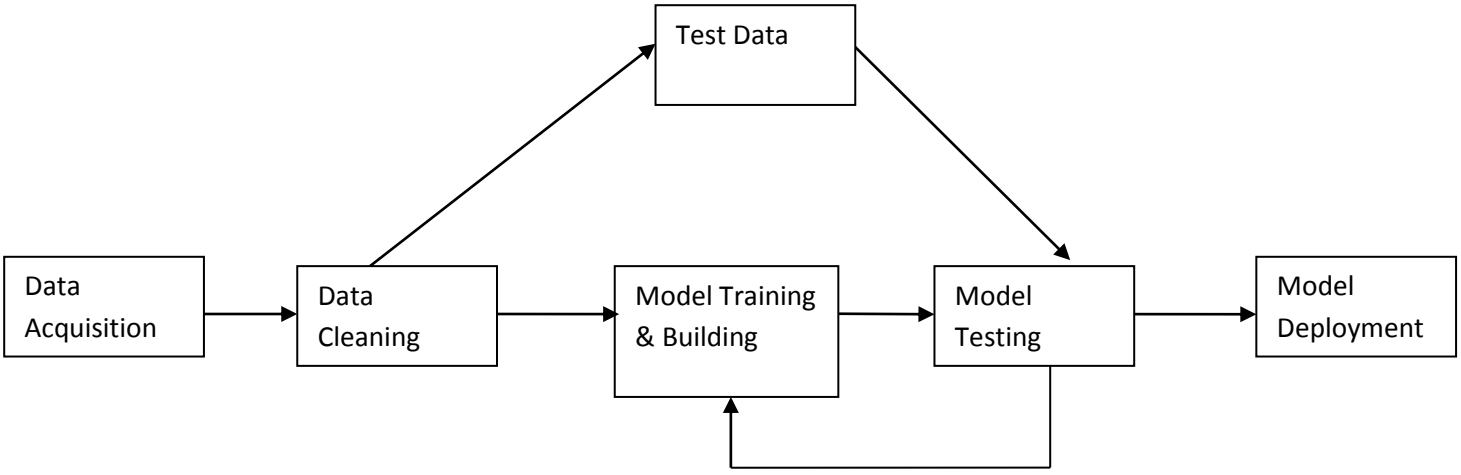


Figure 11.Machine learning Processes

Figure 11 shows six different stages involved in analyzing the measured values of the parameters necessary for determining the drug efficacy that are considered in this project. Data acquisition is the first stage in a machine learning process. It involves the acquiring of data that will be used for training and testing of the model. After this stage, the data is cleaned by ensuring that no missing attribute is present in it. The cleaned data is divided into test and training data. The training data is used to train the machine learning algorithm (Support vector machine in this

case) thereby by building a model that can be tested with the test data. This model can be tested for multiple numbers of times without affecting the result of the trained model. When the testing is completed, the model can then be deployed. These processes are explained in details below.

4.1.1 Data Acquisition

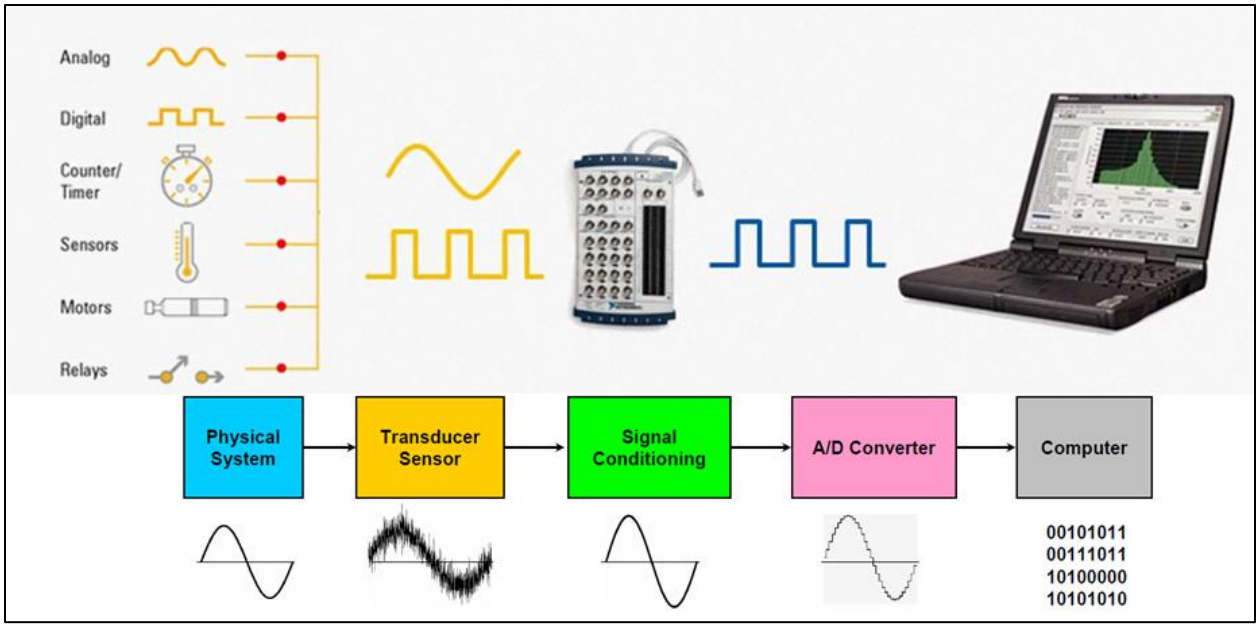


Figure12. Data Acquisition System. (Rroballo 2013).

Figure 12 demonstrate a clearer view of the processes involved in acquiring data using iDRX Series Serial Port data acquisition system or UWTC wireless thermocouple connector system. The physical system consists of pressure, motion, flow that is converted to an electrical form using a transducer. The signal obtained from the transducer sensor is noisy and have to be filtered before the hard ware converts them to a digital form. Updating this data into a particular software (for example matlab or r project for statistical computing) and applying machine learning algorithm; they can be classified or used to predict a specific output. (Jay 2011).

In this project, the values of the parameters used for testing drugs were compiled and randomly generated in order to create a dataset. The data set consist of 10,000 sample cell measurements of

different gender and age groups having defined parameters such as; Solubility (P1), Drug partitioning into different organs (P2), Iron level in blood stream (P3), Zinc in blood stream (P4), Magnesium in blood stream(P5), Unbound fraction of drug (P6) and extraction ratio (P7) that are suitable for determining drug efficacy.

4.1.2 Data Cleaning

This stage of machine learning process ensures that no missing attribute is present in the columns or rows of the data set. The data set can also be filtered or modified as desired in this stage. After this stage the data is divided into test and training data sets.

4.1.3 Training Data Set

Considering 'x' to be an input and 'y' as the output, the training set is $(x_1, y_1), \dots, (x_m, y_m)$. Support vector machine learning algorithm uses a known $x \in X$ in order to find an output $y \in Y$. It aims in studying a classifier of y = f(x, α), α being the parameters of the function. Thus given a data (x) with distinct parameters involved in treating a disease (α), it is possible to find a function f(x,α) that can determine the possibility of that drug being able to treat the disease or not(y). The output can be learned by choosing a function that performs well on the training data thereby minimizing the risk of wrong predictions. (Jason2007.)

The training set = $(x_1, y_1), \dots, (x_m, y_m) \in X$x $\{\pm 1\}$.

The test set = $\overline{x_1}, \dots, \overline{x_m} \in X$

These two sets of functions are arranged in this format so that no interaction should occur between the training and the test set. For any f, there exists f * as shown below.

f*(x_i) = f(x_i) for all i

f*(x_j) ≠f(x_j) for all j

This further reduces the interactions as well. Placing some restrictions on the functions will help to compensate for risks. The true risk involved in a model is a combination of the empirical risk (training error) and an additional term.

(4.3) $$R(\alpha) \leq \left(\frac{1}{M} \sum_{i=1}^{m} l(f(x_i, \alpha), y_i) + \sqrt{\frac{h\left(log\left(\frac{2m}{h}+1\right) - log\left(\frac{\eta}{4}\right)\right)}{m}} \right)$$

Where

(x_i, y_i) are the input and output of the training data (training sets)

m is the number of training sets

α is the parameter of the function

l is the zero-loss function, $l(y, \hat{y}) = 1$ if y ≠ $\hat{y}$, and 0 otherwise.

h is the Vapnik & Chervonenk is (VC) dimension. VC dimension is the maximum point that can be separated in a set of function. It measures the capacity and the complexity of the function.

The overall error in the model should be less than the errors due to the training and model complexity. The higher the capacity of the function (complex model), the lower the training error but it might result to the model being over fitted whereas an increased in the training error is realized in simple set of models. Therefore, keeping the capacity moderate is of great importance. (Jason 2007.)

The training data are usually the values of the parameters necessary for classifying a particular circumstance and their expected output as defined after manufacturing the product to be tested. In this project, different percentages of the data set with their respective cells reactions are used for training the model. This will establish a relation between the amount of data used for training

versus that used in testing the model and their respective classification error obtained in the model. C-classification is the type of SVM that is trained to build this model. This type of SVM enables the error in the model to be minimized in the training data. Using a radial kernel function, the non-linear data are mapped into a high dimensional feature space where a linear model is constructed.

(4.4) $$f(x,w) = \sum_{j=1}^{m} w_j g(x)_j + b$$

Where

$g(x)_j$ is a set of nonlinear transformation

w_j is the slope

b is the bias term.

Equation 4.4 is a mathematically representation of a linear model in a feature space. These data are assumed to be of zero mean thereby making the bias to drop using C-classification SVM type classifier as shown in equation 4.1, the model can be trained.

4.1.4 Separable and Non Separable Data

The sets of hyperplanes for a linear support vector machine (f(x) = (w*x) + b) needs a function that is capable of minimizing the training error and the complexity of the model. This can be written mathematically as: (Jason 2007.)

(4.5) $$\frac{1}{M}\sum_{i=1}^{m} l(f(w * x_i + b), y_i) + ||w||^2$$

Subjected to $min_i|w * x_i| = 1$.

The model complexity is minimized in the case of separable data. ($||w||^2$ is minimized such that $y_i(w * x_i + b) \geq 1$.

With the non-separable case, the model complexity and the parameter in charge of the non-separable data are minimized ($||w||^2 + C \sum_{i=1}^{m} A_i$) in such a manner that $y_i(w * x_i + b) \geq 1$- A_i and $A_i \geq 0$.

The data in a non-linear support vector machine ($w * \phi(x) + b$) needs to be mapped into a higher dimensional feature in order for the data to be classified.

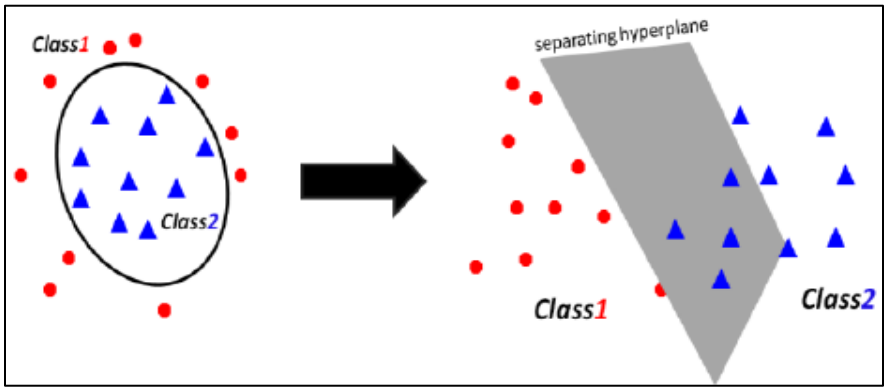


Figure13. Non-linear SVM mapping data from input space into a higher dimensional space (Davood, Marvi, Mehdi & Marzieh).

In figure 13 above, the function ϕ maps the data into a higher dimensional space then construct a hyper plane in that space in such a way that the equations will remain the same. The separating hyper plane is calculated by maximizing the distance between the two classes. (Brandon 2011.)

A decision function determines whether a point is on one side of hyper plane or on the other side. The labeled data and coefficient α are used to teach the decision function. (Boris et al 2010.)

4.6 $f(x) = sign(\sum_{i=1}^{l} \alpha_i y_i x_i x + b)$

4.7 $f(x) = sign(\sum_{i=1}^{l} \alpha_i \, y_i k(x_i x) + b)$

Where

(x_i, y_i) are labeled data

α_i is coefficient

b is a constant

$k(x_i x)$ Defines the kernel functions

Equation 4.6 is a decision function for linear separable data while 4.7 is a decision function for non-linear separable data. The values of x_i for which the coefficient α_i is not zero are known as the support vector. Support vectors are used to separate the hyper plane. Choosing a correct kernel function (Polynomial, radial basis, nonlinear or sigmoid), the data can be transformed, classified and retransform back to the original space. In order to reduce the noise from the data, a penalization parameter (C) is introduced in the SVM's formula as shown below. (Boris et al. 2010.)

(4.8) $k \leftarrow k + \frac{1}{C} I$

I is the identity matrix and C represents the gain realized between the margin maximization and class separation. A higher classification rate requires more noise compensation (Boris et al 2007).

4.1.5 Test Data.

The test data are the required data whose outputs are predicted from the trained model. In this project, the corresponding percentages from the data set remaining after the training data was extracted are used as the test data. Since this test data contains the cells reactions that are to be predicted by the model, when testing the model these cells reactions are excluded from the test data in order to test the effectiveness of the model. Model testing can be done million of times without affecting the manner in which the model is trained or the result of the trained model. Tuning the parameters of the C-classification (cost, kernel parameter and gamma) to a closer range of observed default parameter can estimate the SVM parameters that provide the best model performance. These parameters can then be used in the SVM classifier to train the model thereby providing higher accuracy to the classified output.

4.2 Benefits of Using Support Vector Machine

The fact that SVM operates by maximizing the hyperplanes margin, the studied model can be classified or predicted correctly because the distance from the hyperplane to the nearest pattern (data point) is very wide enough to be classified correctly. This has made SVM to be one of the most effective machine learning algorithms having the following benefits; firstly SVM machine learning algorithm can be used in pattern recognition (speech, face and image recognition). Secondly, there is no need to linearly separate data before applying SVM techniques because there is a presence of a kernel function that can be use to map the non-linear data into a high dimensional feature space. Furthermore, selecting suitable parameters (cost and gamma values) can result in a better performance which enables the prediction of new samples. In addition, SVM produces a unique solution when the optimal solution is rounded unlike neural networks which produce many solutions just with a local minimum thereby making the model not to be trusted over different samples. Lastly, the selection of a good kernel function can provide

similarities between similar operational companies. As the company operations become more similar, the value of the kernel becomes higher. Therefore, in classifying a new company, the financial ratio will be compared to the support vectors. (Sasan et al 2014.)

4.3 Support Vector Machine Challenges

Despite the broad margins and the benefits involved in using support vector machine, some challenges are still realized with this machine learning algorithm. These challenges include the following; firstly, the data used in SVM may not be balanced since some classes have few data while others have more. Secondly, there are multi-label classifications in such a way that an instance might be associated to more than two labels thereby leading to many levels in a factor which could be hard to classify. Furthermore, SVM is not accurate in classifying semi-supervised machine learning problems since some data are left without label and the SVM algorithm only used the labeled data to predict future values or class without considering the unlabeled data. Lastly, SVM finds difficulties in handling larger data scale unless the data are approximated or a linear SVM is used in solving multiclass large data set problems. (Lin 2006.)

4.4 A Simulation to Train a Model Using Support Vector Machine Learning Technique

The objective of this simulation is to train a model that is capable of classifying the reactions of cells to a drug based on different measurements from human cells in a chip. 10,000 measured human cells are used to train and test this model. After the drug is administered in the chip, these cells are expected to react by dying (class 0), having no effect (class 1), responding negatively (class 2) or responding positively (class 3). Based on this response of the measured cells in the chip, the drug can be identified as being effective or not in treating a particular disease. This

simulation is divided into data visualization, training of the model, testing of the model, tuning of the model parameters, and data analysis.

4.4.1 Data Visualization

A dataset consisting of 10,000 measured cells of humans from different age and gender was used in this analysis. It constitutes of 10,000 rows (samples) and 10 columns (information about cells measurement) the columns are: Gender (1 = male, -1 = female), Age, P1, P2, P3, P4, P5, P6, P7, Output. This is listed for every sample/measured human cell. The gender and age parameters in this simulation were used to verify whether the gender group or age of a patient whose cells were measured had an effect with the parameters necessary for determining the drug efficacy (P1, P2, P3, P4, P5, P6 and P7). For a drug to be suitable for treating a disease, it must satisfy the following requirement;

1. Solubility

It is one of the most important drug test requirement which studies the possibility of the drug ability to be absorbed by the body. This possibility of drug absorption is measured with the potential of hydrogen (pH). pH is the measure of the protons concentration in a solution, it gives an information about the acidic or alkalinity of a substance. Substances with pH between 0 - 7 are considered to be acidic why those between 7-14 are alkaline. The required pH for the drug to be considered soluble is between 1- 7.5. Therefore, the drugs have to be highly acidic or slightly alkaline (below 8 pH) for it to be considered soluble. This is referred to as “P1” in the data set.

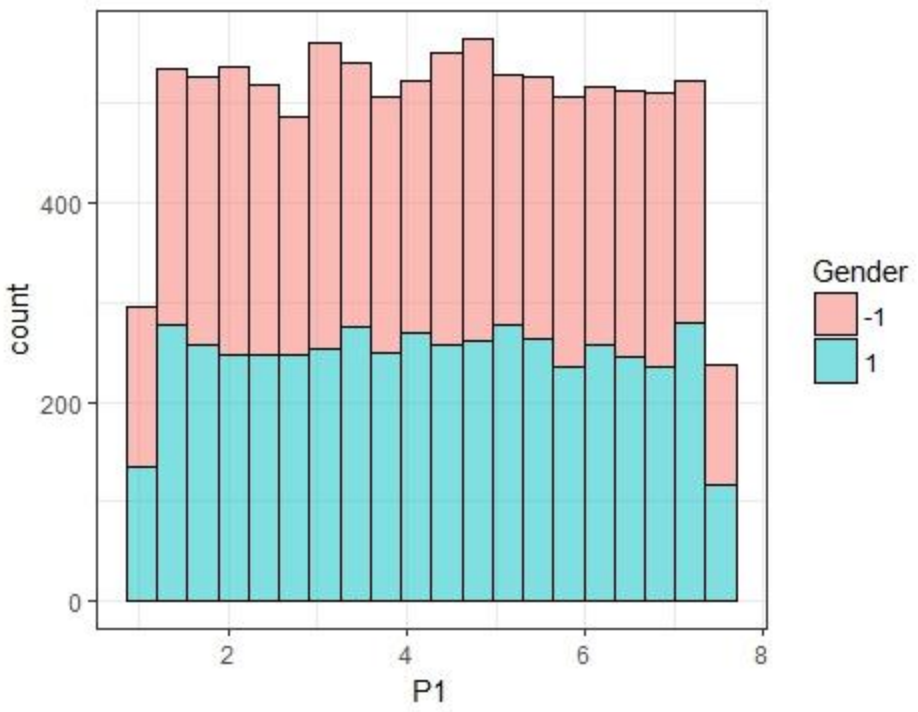


Figure14. Histogram of drug solubility

The histogram above represents the solubility range of the data. Figure 14 indicates that the required pH level (1 to 7.5) was used for the data analysis with the same level of solubility being observed for both male and female at every range. The pH levels between 3 - 3.5 and 5 -5.1 were recorded to be more frequent between the male and female. Hence the nature of the drug was mostly acidic to both male and female.

2 Drug partitioning into different Organs

This is another important parameter in testing drug efficacy. It determines the partitioning ratio of the tissue to blood in order to describe the distribution of chemicals in the body. This distribution of drugs to the body is defined as the drugs leave the blood stream to enter the extracellular fluid and cells/tissues. Poorly perfused tissues or cells such as muscle or fat have a very slow fluid to tissue flow hence a low partitioning coefficient while highly perfuse organs

such as liver, heart and kidney experience a high distribution drug due to the fast flow of fluids over these organs, thus a higher partitioning coefficient is experienced when the drugs reach into this organ. For a drug to be partitioned correctly, the partitioning coefficient should be greater than 100,000 whether with the slow or fast perfuse organs. (Howmed 2011.)
This can be represented mathematically as follows:

$\log(p) > 5$.

'p' is referred to as the partition coefficient.
This parameter is denoted "P2" in the data set.

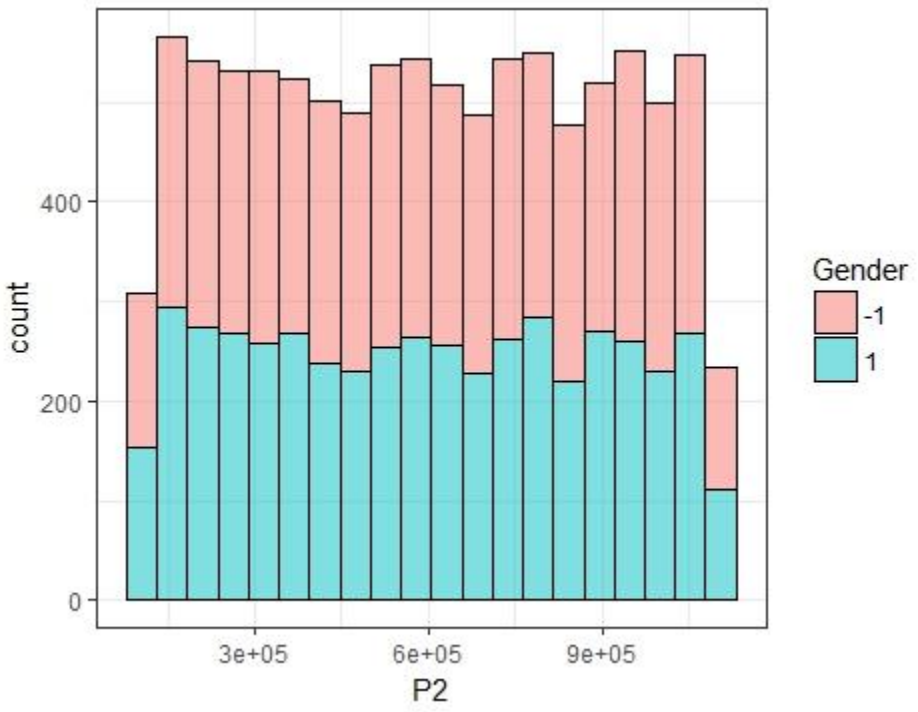


Figure15. Histogram for drug partition

Figure15 above shows the range of partition coefficients observed in data. The smallest count in the drug partition is observed at p = 1,200,000 and the highest count is when p = 200,000. The

most frequently recorded partition coefficient was around 300,000. This partitioning coefficient is equally distributed for both male and female throughout the entire range from 100,000 to 1200,000.

3. Iron Level in the blood stream

This is the required plasma concentration of drug that is obtained for Iron. A reduction in Iron level is observed in a sick person cell but upon taking treatment, this iron level will begin to increase. For example, the use of Cisplatin by cancer patients improves their iron level to be 0.6 to 1.9 g/l. This change in iron level was realized during the early stages of administering Cisplatin. Thereby indicating that iron level in the blood stream can be able to determine the response to cancer treatment with Cisplatin. (Biomed 2016.)

This parameter is denoted as ‘P3’ in the data set

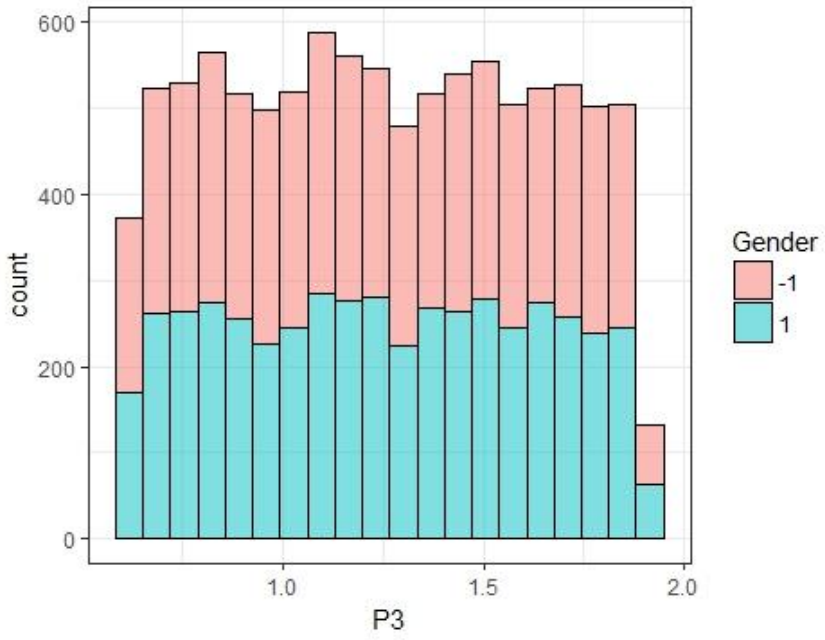


Figure 16. Histogram for Iron Level.

From figure16, it is seen that the iron level is equally observed between male and female in the data set. 12% of the data set has an iron level of 1.9, while 98 % of the data has an iron level of 1.2.

4. Zinc level in the blood stream

This is the required plasma concentration of drug that is obtained for zinc. Just as in iron, zinc level in the blood stream increases during the early days of cancer treatment with Cisplatin. Thus, zinc is also very useful in determining the response to the cisplatin cancer treatment. If patient's zinc level reaction in a measured cell is 0.3 to 3.9g/l then the treatment is actually working. (Biomed 2016.)

This parameter is denoted as "P4" in the data set.

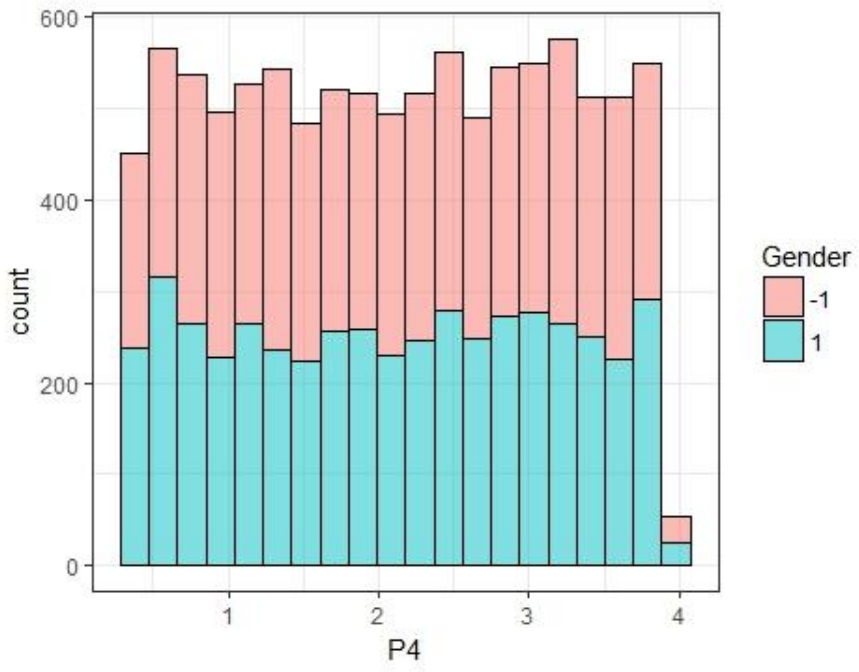


Figure17. Histogram for the Zinc level

As shown in figure 17. The Zinc level in the data set ranges from 0.3 to 4. The smallest count of zinc level is noticed above 4g/l while the highest count of the zinc level is between 0.5 – 0.6g/l.

5. Magnesium level in the blood stream

Magnesium is needed to produce energy, oxidation phosphorylation and glycolysis to the body. Thus, a patient recovering from cancer treatment by cisplatin will need to prove that his/her level of plasma concentration of drug that is obtained for Magnesium is high. The required Magnesium level for a patient's cell recovering from cancer using Cisplatin drug is 0.1 to 5.5gl. (Biomed 2016.)

This parameter is denoted as 'P5' in the data set.

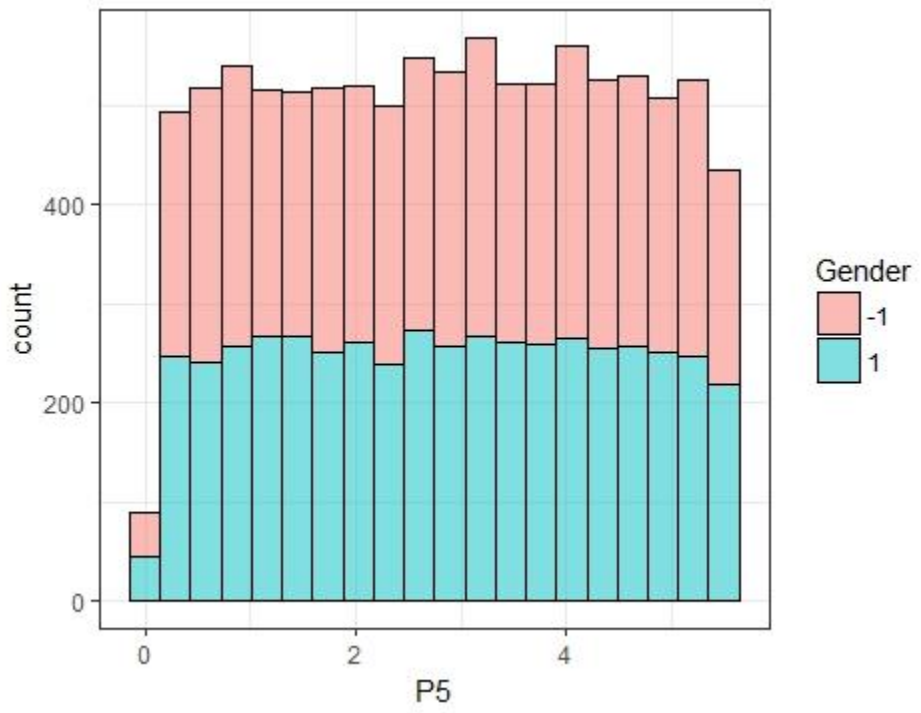


Figure18. Histogram of Magnesium level in the blood

Figure18 shows the ranges and count of magnesium level in the data set. It is equally distributed between male and female and ranges from 0 to 5g/l. The lowest record of the level of magnesium is 0 g/l will the highest count is between 3.1 to 3.4 g/l.

6. Unbound fraction of drug

This is the proportion of drug that is free to penetrate to the surrounding tissues. It affects the time the drugs stay in the body and it is this portion of the drug that have a biological effect or that can be excreted. The bound fraction of the drug on the other hand serves as a reservoir for drug removal which in turn prolong the duration of drug reaction in the body. Therefore, the more the drug is unbound to the plasma protein, the greater its reactive to the body and thus, a better efficacy can be obtained. The smallest allowed proportion of the drug to penetrate to the surrounding is 0.1 while the greatest proportion is more than 0.4. (Asteris 2017.)
This parameter is referred to as P6 in the data set

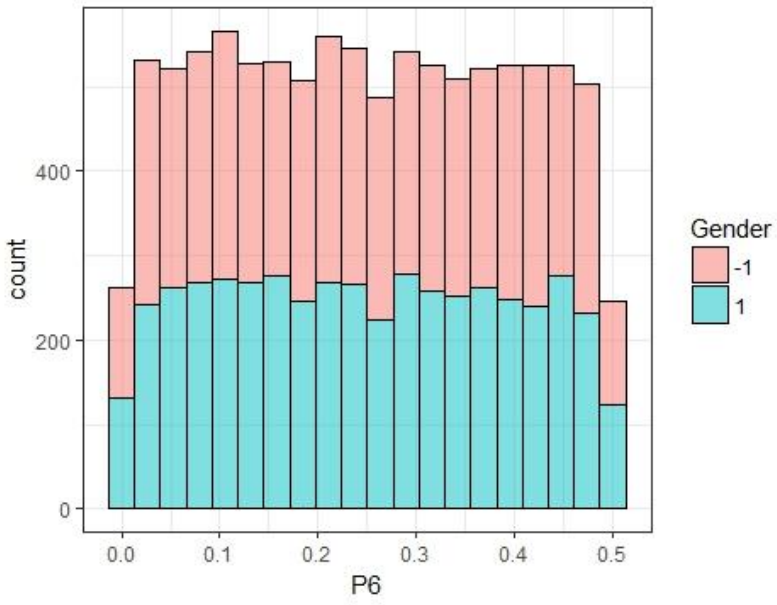


Figure18. Histogram of Unbound fraction of drug

From figure 18, it shows that the unbound fraction of drug distributed in this data is from 0.0 to 0.5 with equal amount realized at each range for both male and female.

7. Extraction Ratio.

This is the ratio of the hepatic clearance of a drug to the hepatic blood. It is the amount of blood in the liver that is cleaned by the drug within a period of time. The blood flowing through the liver, the drug fraction that is not bounded to the plasma protein and the possibility of the hepatic enzymes to maintain the drug, are the three factors that determine the extraction ratio. When this ratio is greater than 0.7 it is referred to as high, when the ratio is between 0.3 to 0.7 it is described as 'intermediate' and when it is less than 0.3 it is low. Therefore, the correct range for extraction ratio is 0.3 to 0.7. (Tusom 2013.)
This parameter is denoted as P7 in the data set.

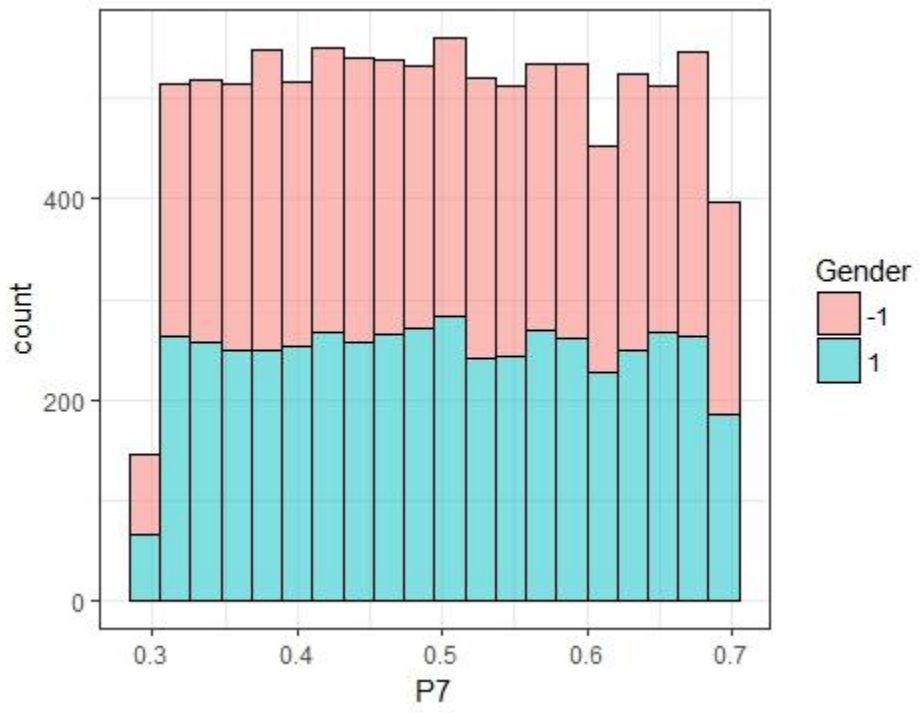


Figure19. Histogram for the extraction range.

Figure 19 shows that the extraction ratio used in the data was from 0.3 to 0.7 with the smallest amount of extraction ratio being 0.3 while the largest count recorded is at 0.5.

In order to measure these parameters, the drug have to be inserted in to the chips then allowed to stay for few days before scanning the cells to measure the values for the parameters necessary for determining drug efficacy. This will ensure the penetration of drugs into the respective organs and also provide time for metals (iron, zinc and magnesium) levels in the blood stream to response to the drug treatment. As a conclusion from the histograms, it is realized that the gender of the patient does not influence the parameters required to determine the drug efficacy since all the parameter ranges are the same for both male and female.

Cell response with respect to the gender of the patient

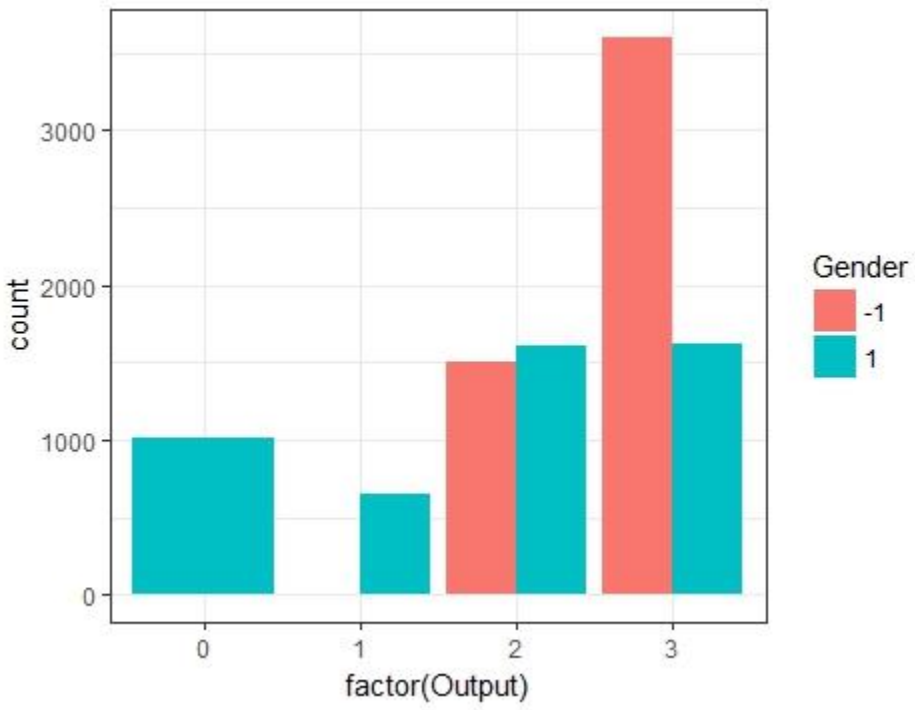


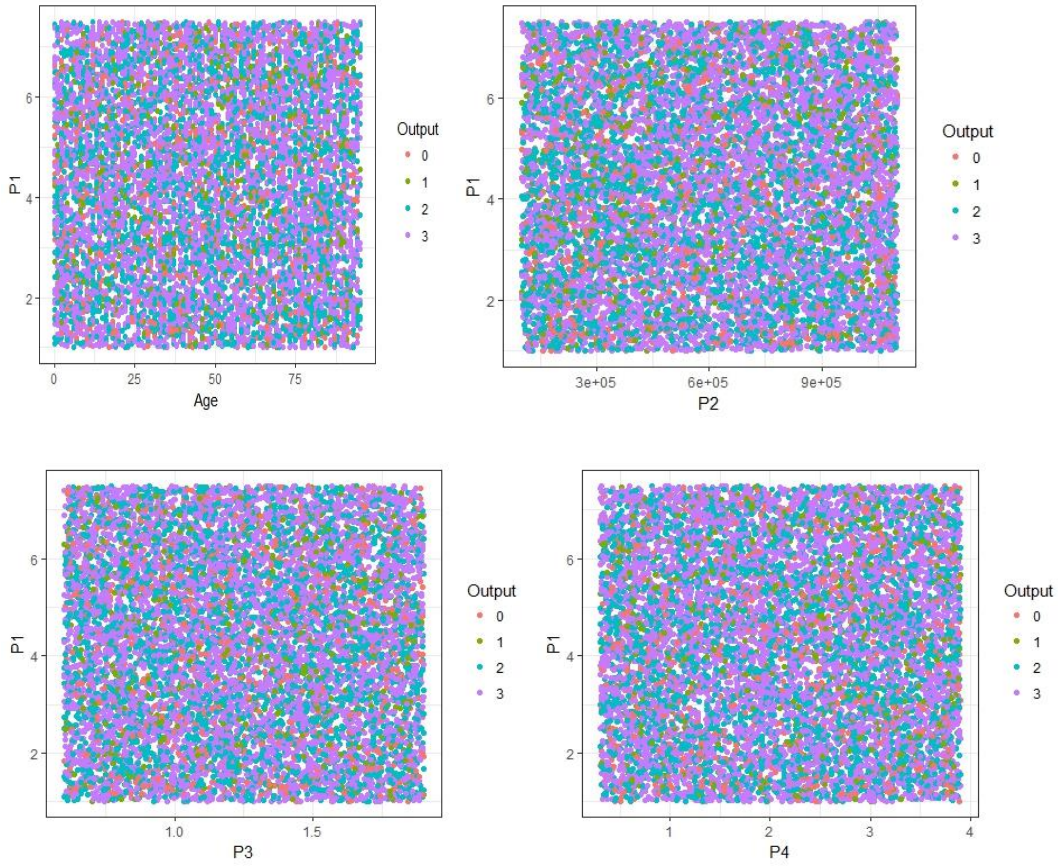
Figure 20. Cell response with respect to the gender of the patients

Though the male and female gender groups do not have an influence on the measured parameters used to determine the drug efficacy, the actual response of these cells to the drug is influenced by the patient gender in which the cell was extracted. In general cells from the male patients have bad reactions. This can be seen from figure 20 in which the dead and unaffected cell responses

are only seen in the cells from male patients as well as the cells from the male patients having the highest negative influence on the cell reaction while the cells from the female patients have the highest positive reactions.

4.4.2 Training the Model

If an oral drug is not absorbable, then it cannot penetrate into a human body. This has made the drug solubility test to be the most important parameter in checking for efficacy of developed oral drugs. Due to this effect, a scattered plot is drawn for all the parameters necessary for determining the drug efficacy in order to verify the relationship between these parameters and the drug solubility based on the cells reactions.



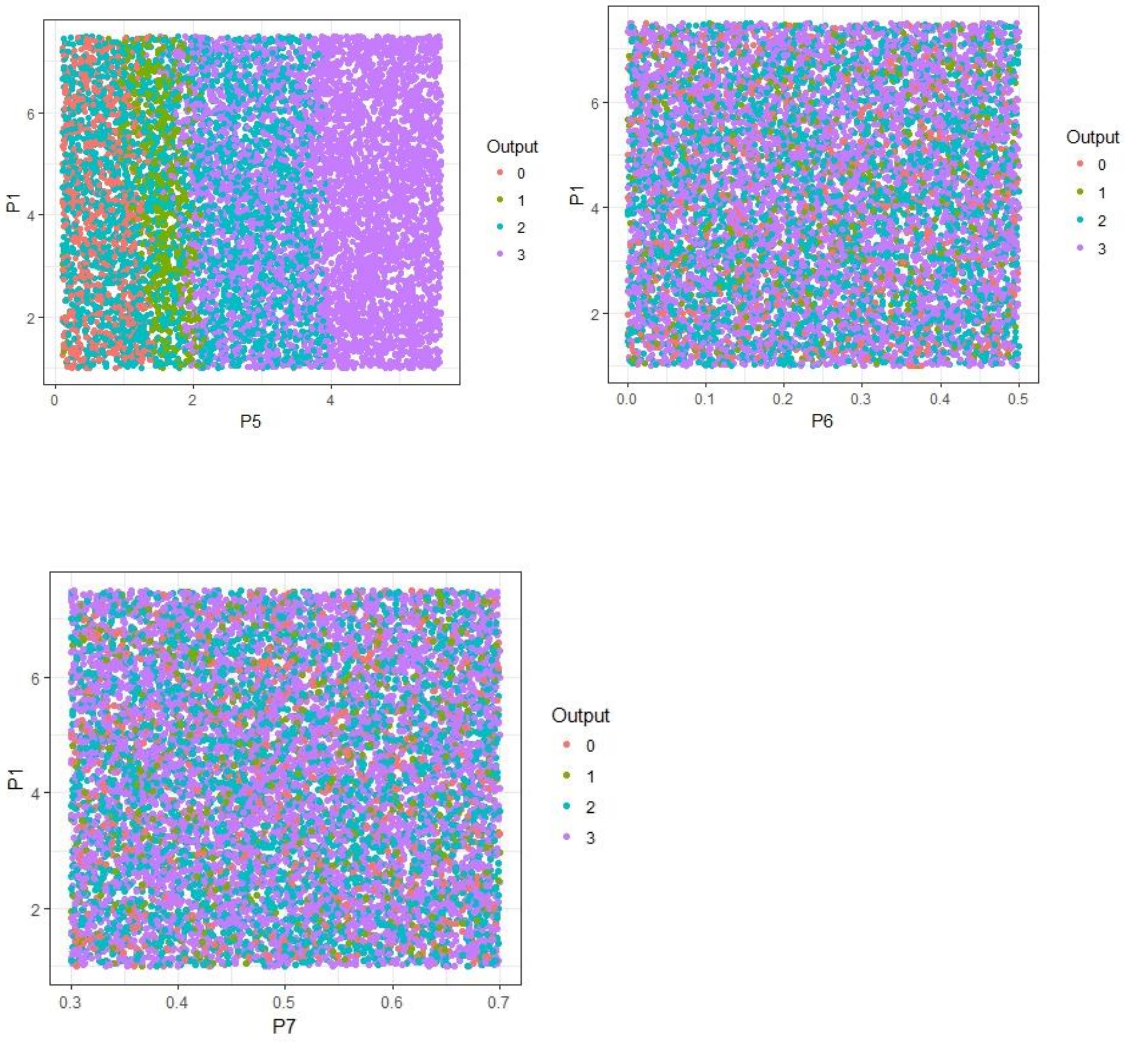


Figure21. Scattered plots showing the relationship between drug solubility and other parameters for drug efficacy.

From the scattered plots above, it is clear that dead cells (orange), unaffected cells (green), negative reactions of cells (blue) and positive reactions of cells (purple) can never be separated linearly and they require a kernel function to map them to a higher dimension before their separation. Also, since it is a medical data with few numbers of classes, greater accuracy is required and support vector machine learning techniques are well known to have different kernel functions (radial, polynomial, linear) that can map these non-linear data into a high dimensional feature space where a linear model can be constructed and trained with accuracy. This is the reason why support vector machine is employed to train the model in this simulation.

This model therefore aims at predicting the cells response (Output) based on the measured parameters from the cells which will enable similar data set to be classified as dead, unaffected, negative response or positive response when tested with the model.

The data was spliced into test and training data. Using 90% of the data sample for training and 10% for testing, a model is trained to classify the cell response (Output) into four levels (0, 1,2 and 3) based on the parameters necessary for determining the drug efficacy.

4.4.3 Summary of the Trained Model

The following summary is obtained from the trained model with 9000 samples

```
Call:
svm(formula = Output ~ ., data = train)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.1

Number of Support Vectors:  2619

 (652 339 1084 544)


Number of Classes:  4
```

```
Levels:
 0 1 2 3
```

This confirms the fact that the SVM-type used in this model is a C-classification having 2619 support vectors with four levels of classes. This model uses a radial kernel function to map the non-linear data to a higher dimensional feature space where a linear model is constructed and used to train the model. The penalization parameter (cost = 1) which controls the influence of the support vector thereby minimizing error in the model. Gamma on the other hand measures the similarity between two points. The value of gamma= 0.1 in this model which signifies that each training sample was taken into consideration when calculating the decision boundary thereby producing a more linear function while high gamma turns to yield weak decision boundaries.

In order to ensure that this model was correctly trained, the test data was used for testing the cell reactions. This testing was done by omitting the output column from the test data during the prediction.

This yielded the following output

```
predict_values     0   1   2   3
                 0 86   5   0   0
                 1  5  58   5   0
                 2  0  11 291  14
                 3  0   0   8 517
```

The table used to arrange this predicted output is known as the confusion matrix. The confusion matrix contains the total number of counts that an output in the model was correctly classified or not. These counts are grouped under True Positive, True Negative, False Positive and False Negative for each classification level

True positives; these counts represents the total instances that were correctly classified (labeled) in a given class. In the confusion matrix, the true positives are the diagonal of the matrix.

True negatives; these are the total non-instances in a class that were correctly labeled by the classifier (the Outputs that were correctly identified to be in different classes when considering a particular level of the classifier).

False positives; these are the total non-instances in a class that were incorrectly labeled as being in that class (The outputs that were incorrectly classified by the classifier as belonging to the class of interest. These counts are found on the column of the class of interest expect the entry of that specific class.)

False negatives; these are the total instances in a class that were mislabeled as not being in the class. (The total false negative of a class is the sum of the class output entries in the row of that specific class apart from the entry in that class)

Therefore, this model was able to clearly classify 86 out of 1000 cells as being dead (True Positive) while 5 of the cells were wrongly classified as not being dead by the classifier (False Negative) whereas the classifier falsely classified 5 cells that belong to unaffected cells as being dead cells (False Positive). This interpretation is applicable for 1, 2 and 3 class levels.

By comparing the predicted output with the actual out, the specific wrongly classified cells can be known.

Only the first six outputs were compared because of the bulkiness of sample test data.

```
  predicted actual
14        4      4
22        4      4
25        3      3
36        4      4
40        4      4
48        3      3
```

These six outputs happened to be predicted correctly. If the entire results were printed, the gender/age where the cells reactions were wrongly classified could have been identified.

In order to improve these results, the SVM classifier parameters are tuned to check the best parameter values that can be used to train and test the model. The summary for this tuning is shown below;

```
Parameter tuning of ‘svm’:
.
- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  1.5   0.1

- best performance: 0.03155556

- Detailed performance results:
  cost gamma      error  dispersion
1  0.5   0.1 0.03744444 0.006688237
2  1.0   0.1 0.03511111 0.007223647
3  1.5   0.1 0.03155556 0.005137345
4  0.5   0.5 0.08977778 0.011674600
5  1.0   0.5 0.08355556 0.010512257
6  1.5   0.5 0.08188889 0.008115253
7  0.5   0.7 0.11266667 0.015018964
8  1.0   0.7 0.09577778 0.010394153
9  1.5   0.7 0.09511111 0.010438927
```

This summary of the tuned parameters shows that 0.1, 1 and 1.5 cost values with the gamma values of 0.1, 0.5 and 0.7 were tuned in the model in order to choose the best SVM parameter that will be used to train the model. According to this summary, the best performance was obtained with a cost of 1.5 and gamma of 0.1. After the best parameters for the model are being tuned, they are used to test and classify the output.

Output classification with these parameters gave the following results.

```
tuned predictions    0   1   2   3
                  0 87   3   0   0
                  1   4 61   4   0
                  2   0  10 293   9
                  3   0   0   7 522
```

This is an improved classification when compared to the one in which the parameters were still default without being tuned. For example, they are 87 out of 1000 cells classified as dead (True Positive) while 3 of the cells types were wrongly classified as not being dead by the classifier (False Negative) whereas the classifier falsely classified 4 cells types that were unaffected to be dead cells (False Positive) as opposed to 86 correctly classified cell types, 5 false negative and 5 false positive in level 0 seen when the parameters were not yet tuned.

In order to verify the Output classification when the gamma parameter was increased to 0.5, another model was trained with a gamma of 0.5 and the cost parameter was kept constant. Testing the sample data with these SVM parameters yielded the following output.

```
tuned_predictions1     0   1   2   3
                    0 79  11   0   0
                    1 11  41   7   0
                    2  1  22 278  14
                    3  0   0  19 517
```

It can be seen that the levels of classification were worst in this case when compared to the level of classification at the time when the SVM parameters were not yet tuned.

4.4.4 Analysis.

Taking the three models in to consideration, their accuracy can be achieved as follows;

$$ModelAccuracy = \frac{correctlyclassifiedsample}{totaltestedsample} x\ 100\%$$

Accuracy of the Model before tuning

$$\text{Model \_ Accuracy \_before\_ tuning} = \frac{952}{1000} x\ 100\%$$

Model _ Accuracy _before_ tuning = 95.2%

Accuracy of the model with cost = 1.5 and gamma = 0.1

Model_Accuracy1 = $\frac{963}{1000} x100\%$

Model_Accuracy1 = 96.3%

Accuracy of the model with cost = 1.5 and gamma = 0.5

Model_Accuracy2 = $\frac{915}{1000}\ x\ 100\%$

Model_Accuracy2 = 91.5%

The above calculations shows that the best accuracy of a trained model is obtained at a higher cost and a lower gamma parameter of the SVM classifier which ensures that the support vectors have a high influence of the training data and produces a stronger decision boundary respectively while lower cost and high gamma parameters will results to a poorly classified model since the support vectors have low influence on the training data sample and the decision boundary between the Output and other parameters are weak.

In order to determine the sensitivity (ability to identify the correct reactions in each class) and specificity (ability of the model to identify faulty reactions in a class) of the most accurate model, the True Positives, True Negatives, False Positives and False Negatives of each class need to be determined.

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Specificity = \frac{TrueNegative}{FalsePositive + TrueNegative}$$

Sensitivity and specificity of the respective classes from the best model

Tp = True Positive

Tn = True Negative

Fp = False Positive

Fn = False Negative

Sensitivity and selectivity for dead cell (class 0)

Tp_0 = 87

Tn_0 = 876

Fp_0 = 4

Fn_0 = 3

Sentivity_0 = (Tp_0/(Tp_0 + Fn_0)) * 100

Specificy_0 = (Tn_0/(Fp_0 + Tn_0)) * 100

Sentivity_0 = 96.66667

Specificy_0 = 99.54545

Sensitivity and selectivity for the cells unaffected (class 1)

Tp_1 = 61

Tn_1 = 902

Fp_1 = 13

Fn_1 = 8

Sentivity_1 = (Tp_1/(Tp_1 + Fn_1)) * 100

Specificy_1 = (Tn_1/(Fp_1 + Tn_1)) * 100

Sentivity_1 = 88.4058

Specificy_1 = 98.57923

Sensitivity and specificity for cells with negative reaction (class 2)

Tp_2 = 293

Tn_2 = 670

Fp_2 = 11

Fn_2 = 19

Sentivity_2 = (Tp_2/(Tp_2 + Fn_2)) * 100

Specificy_2 = (Tn_2/(Fp_2 + Tn_2)) * 100

Sentivity_2 = 93.91026

Specificy_2 = 98.3847

Sensitivity and specificity for cells with positive reactions (class 3)

Tp_3 = 522

Tn_3 = 441

Fp_3 = 9

Fn_3 = 7

Sentivity_3 = (Tp_3/(Tp_3 + Fn_3)) * 100

Specificy_3 = (Tn_3/(Fp_3 + Tn_3)) * 100

Sentivity_3 = 98.67675

Specificy_3 =98

Therefore, this model identified the positive response on cells (class 3) as being highly sensitive and dead cells as the most specified cell reactions.

In order to verify whether the sample size of the training data as well as that of the test data can have an influence on the model classification, 5%, 10%, 20%, 30%,40%,50%,60%,70% and 80% of the data set is used in training the model while the corresponding data percentages is used as

test samples. After the data is trained, it is tested in order to predict and classify the cell response to the drug. Comparing the predicted output with the actual output, the correctly identified outputs are obtained and used in calculating the errors involved in a classification (Classification error).

$$Classification_{error} = 1 - \frac{sum(CorrectlyIdentifiedOutput)}{TotalPredictedOutputs}$$

The classification error for using 5% of the data set as the training data is;

ClassificationError5 $=1 - \frac{8363}{9499}$

ClassificationError5 = 0.1195915

The classification error for using 10% of the data set as the training data is;

ClassificationError10 $=1 - \frac{8199}{9000}$

ClassificationError10 = 0.089

The classification error for using 20% of the data set as the training data is;

ClassificationError20 = $1 - \frac{7503}{8001}$

ClassificationError20 = 0.06224222

The classification error for using 30% of the data set as the training data is;

ClassificationError30 = $1 - \frac{6646}{7000}$

ClassificationError30 = 0.05057143

The classification error for using 40% of the data set as the training data is;

ClassificationError40 $= 1 - \frac{5721}{6001}$

ClassificationError40 = 0.04665889

The classification error for using 50% of the data set as the training data is;

$$\text{ClassificationError50} = 1 - \frac{4812}{4999}$$

ClassificationError50 = 0.03740748

The classification error for using 60% of the data set as the training data is;

$$\text{ClassificationError60} = 1 - \frac{3862}{3999}$$

ClassificationError60 = 0.03425856

The classification error for using 70% of the data set as the training data is;

$$\text{ClassificationError70} = 1 - \frac{2905}{3000}$$

ClassificationError70 = 0.03166667

The classification error for using 80% of the data set as the training data is;

$$\text{ClassificationError80} = 1 - \frac{1946}{1999}$$

ClassificationError80 = 0.02651326

The classification error for using 90% of the data set as the training data is;

$$\text{ClassificationError90} = 1 - \frac{976}{1000}$$

ClassificationError90 = 0.024

A graph showing the relationship between the classification error and the percentage of data set used in training the model is shown in the figure below;

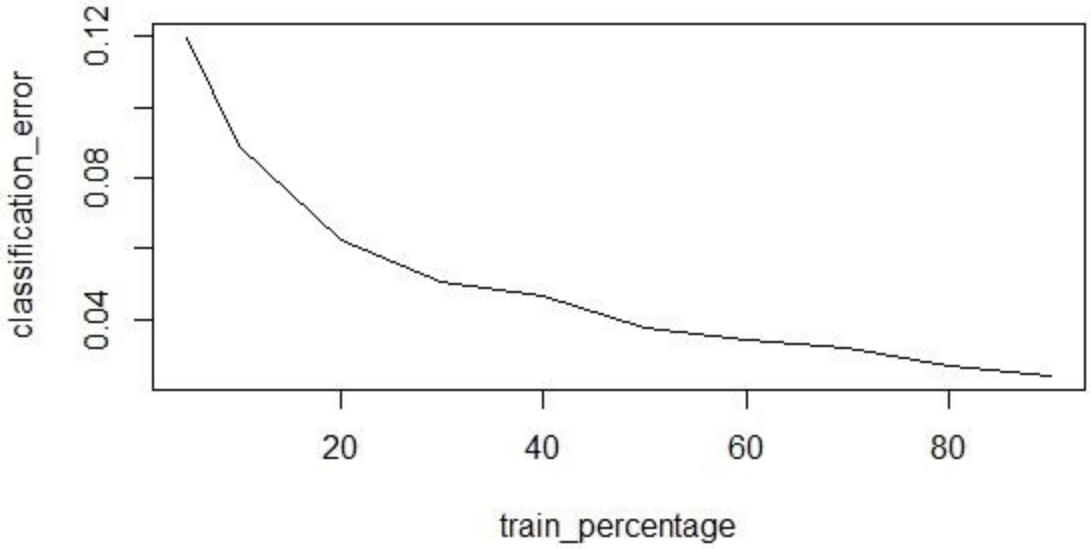


Figure22. The relationship between classification error and the percentage of data used for training

Figure 22 shows that the classification error decreases as the percentage of the data used for training increases. At the smallest percentage of data used for training, the biggest classification error of 0.12 was obtained while when the biggest percentage of data was used for training and a small amount used for testing, the classification error as small as 0.024 was realized.

Therefore, if the parameters necessary for determining the drug efficacy of a developed drug have fewer classes whose values are balanced within each class, the labeled data can be used to train the SVM machine learning algorithm in order to build a model that can be able to predict the cells reactions to the drugs based on the measured parameters from the cells in the chip. When a sample test data is very small as compared with the data that is used in training the model, a smaller classification error will be obtained but even if the test data was bigger than the data used in training the model, a classification error of 0.12 will be obtained which still is a reasonable error that can be easily traced. To improve the classification of the cell response to the drug it is recommended to use a high cost and a low gamma SVM parameter values according to the result of the best performance after the SVM parameters are tuned. SVM

machine learning techniques can thus be reliably used to train and predict the reactions of cells in a chip in order to determine the drug efficacy.

5 CONCLUSION

Organ-on-a-chip devices have been increasingly developed to model various tissues and organs. They can be used to study physiology, pathology and develop therapies. When more than one organ-on-a-chip types are combined, multi organ-on-a-chip devices are obtained. In addition, several organ-on-a-chip devices can be linked together in an array. With sensors linked to these chips to capture various changes, very big data is generated. This imposes a real challenge in analyzing the obtained data and deriving cells reactions. Machine learning is a very good data analysis method that can learn from a trained data set and then test other sample data based on the knowledge from the train data set without being programmed. The machine learning algorithm used for the training and testing the model depends on the data sample that is used in the model. Support vector machine learning technique is used in circumstances when the required training data set has few numbers of classes with these classes having almost the same amount of data. SVM maps non-linear data set to a higher dimensional feature space using a kernel function where a linear model can be constructed and trained. Tuning the parameters of the SVM classifier will result to the selection of these parameters according to the best performance. When these best performance parameters are used in training a model, a higher accuracy as well as a lower classification error will be obtained. In general, the higher the cost parameter of the SVM model with a lower gamma parameter value, the more accurate a model can be classified whereas if the cost value is small and the gamma value is large, the model cannot be classified appropriately. Also the training and test data samples have an influence in the results obtained when the model is being tested. Smaller trained data sample and bigger test sample results to a large classification error while bigger trained data sample with smaller test data samples produces smaller classification error. Smaller classification errors are easy to be traced and corrected. Though SVM is very reliable and effective in predicting medical data due to the fact that their data samples always have few classes (for example most reactions in a medical data is either positive, negative, unaffected or dead), some data set might have more samples in one class than others in some cases thereby making this type of data sample unsuitable to be classified by the SVM machine learning techniques. In this case other suitable machine learning techniques can be used to train and test the data. Thus, depending on the

extracted parameters from the cells in the integrated chip, a particular machine learning algorithm can be trained to predict the cells reactions to the drugs thereby determining the drug efficacy. Therefore organs-on-a-chip in parallel will be able to test the drug efficacy within a reduced time, cost and risk.

References

ABM, (n.d). *An introduction to cell culture*. [Cited 28.12.2017]. Available at: https://www.abmgood.com/marketing/knowledge_base/cell_culture_introduction.php

Alpaydin, Ethem (2014). *Introduction to machine learning*. USA: MIT Press. [Cited 03.02.2018]. Available at: https://ebookcentral-proquest-com.proxy.uwasa.fi/lib/tritonia-ebooks/reader.action?docID=3339851&query=ethem%202014

Arun, Alvappillai & Peter, Neal Barrina (n.d.). *Face recognition using machine learning*. [Cited 01.03.2018]. Available at: http://noiselab.ucsd.edu/ECE285/FinalProjects/Group7.pdf

Bansal (2017). *Regression and classification| supervised machine*. [Cited 05.02.2018]. Available at: https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/

Bell, Jell (2014). *Machine learning: hands-on for developers and technical professionals.*[Cited 23.01.2018]. Available at: https://ebookcentral-proquest-com.proxy.uwasa.fi/lib/tritonia-ebooks/reader.action?docID=1818248&query

Bernd, Klein (2011). *Text categorization and classification*. [Cited 19.02.2018]. Available at: https://www.python-course.eu/text_classification_introduction.php

Bhanu, Prasad (2016). *A review on drug testing in animals*. India: Transl Biome. [Cited 13.01.2018]. Available at: http://www.transbiomedicine.com/translational-biomedicine/a-review-on-drug-testing-in-animals.php?aid=17736

Boris, L. Priscille, B. Hee, Kyoung. Thierry, C & Auguste.G (2007). *Support vector machines for automatic detection of tuberculosis bacteria in confocal microscopy images*. USA: IEEE. [Cited 20.03.2018]. Available at: http://ieeexplore.ieee.org.proxy.uwasa.fi/document/4193228/authors?reload=true

Brandon, Boyle (2011). *Support Vector Machine: Data analysis, machine learning and applications.*USA: Nova Science. [Cited 07.03.2018]. Available at: https://ebookcentral-proquest-com.proxy.uwasa.fi/lib/tritonia-ebooks/reader.action?docID=3021500&query=SVM

Christopher, Prabhakar & Hinrich (2008). *Support vector machines: linearly separable case:* [Cited 12.03.2018]. Available at: https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html

Chih-Jen, L (2006). *Support vector machines: status and challenges.* [Cited 29.03.2018]. Available at: https://www.csie.ntu.edu.tw/~cjlin/talks/caltech.pdf

Davood, M. Hossein, M. Mehdi. T & Marzieh , M (2011). *Age estimation based on speech features and support vector machine*. [Cited 20.03.2018]. Available at:

https://www.researchgate.net/figure/Nonlinear-SVM-Mapping-data-from-input-space-left-into-the-higher-dimensional-feature_fig1_225088888
Elizabeth, Dougherty (2010). *Living, breathing human lung-on-chip.* USA: The Harvard Gazette. [Cited 03.01.2018]. Available at:https://news.harvard.edu/gazette/story/2010/06/living-breathing-human-lung-on-a-chip/

Elinor (2012). *Gut on a chip*. [Cited 30. 12.2017]. Available at: http://prospect.rsc.org/blogs/cw/2012/03/23/gut-on-a-chip/

Harish, Srinivasan. Sargur, Srihari & Matthew Beal. *Machine learning for signature verification*. [Cited 03.04.2018]. Available at: http://www.cedar.buffalo.edu/~srihari/papers/ICGVIP2006-sig.pdf

HowMed, (2011). *Distribution of Drugs.* [Cited 11.04.2018]. Available at: http://howmed.net/pharmacology/distribution-of-drugs/

Gideon, Isabelle & Fishel (2008). *Applications of supervised learning*.
Jason, Weston (n.d.). *Support Vector Machine.* [Cited 17.03.2018]. Available at: http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf

Jason , Brownlee (2016). *Logistic regression for machine learning*. [Cited 07.02.2018]. Available at: https://machinelearningmastery.com/logistic-regression-for-machine-learning/

Jesus, Rodriguez (2017). *Understanding semi-supervised learning.* [Cited 03.03.2018]. Available at: https://medium.com/@jrodthoughts/understanding-semi-supervised-learning-a6437c070c87

Jones, Tim (2017). *Unsupervised learning for data classification*. [Cited 01.02.2018]. Available at: https://www.ibm.com/developerworks/library/cc-unsupervised-learning-data-classification/index.html

Jonathan, Himmelfard & Joseph, Eschbach (2013*). Innovation in kidney disease Research: the Human Kidney on a Chip*. [cited 03.01.2017]. Available at: https://www.nibib.nih.gov/sites/default/files/S2_Himmelfarb.pdf

Kumar, Ajitesh (2015). 7 most common machine learning tasks & related methods. [cited 28.01. 2018]. Available at: https://vitalflux.com/7-common-machine-learning-tasks-related-methods/

Machine Learning: *What it is and why it matters.(*n.d.). [Cited 22.01.2018]. Available at: https://www.sas.com/en_us/insights/analytics/machine-learning.html#

Lawrence X,Yu (2002). *Scientific considerations of polymorphism in ANDAs.* [Cited 02.04.2018]. Available at: http://slideplayer.com/slide/6902148/

Olivier, Chapelle. Bernhard, Scholkopf & Alexander, Zien (2006). Semi-supervised learning. [Cited 04.04.2018]. Available at: https://www.molgen.mpg.de/3659531/MITPress--SemiSupervised-Learning.pdf

Patrick, W. Eun-Young, K. Jeff, T. Tor-Kristian, J. Staal, A. Lucila, O (2004). *A primer on gene expression and microarrays for machine learning researchers*. [Cited 13.02.2018]. Available at: https://www.sciencedirect.com/science/article/pii/S1532046404000693

Roemer, Vlasveld (2013). *Introduction to one-class support vector machines.* [Cited 22.03.2018]. Available at: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/

Rroballo (2013). *Using GeneXus in Data Acquisition and control systems.* [Cited 16.05.2018]. Available at: https://wiki.genexus.com/commwiki/servlet/wiki?20387,Using+GeneXus+in+Data+Acquisition+and+Control+Systems,

Sasan, K. Shahidan M. Mehran, H. Jafar, S & Mohammed, J (2014). *Advantage and drawback of support vector machine functionality*. Malaysia: IEEE. [Cited 27.03.2018]. Available at: https://ieeexplore-ieee-org.proxy.uwasa.fi/document/6914146/

Sean, Henahan (1998). *Stem cell breakthrough*. USA: Access Excellence. [Cited 15.01.2018]. Available at: http://www.accessexcellence.org/WN/SU/stem1198.html

Schmalz, G (1994). *Use of cell cultures for toxicity testing of dental materials –advantages and limitations. G*ermany: University of Regensbury. [Cited 16.01.2018]. Available at: https://www.sciencedirect.com/sdfe/pdf/download/eid/1-s2.0-0300571294900329/first-page-pdf

Scikit- Learn (n.d). *Support vector machines.* [Cited 09.02.2018]. Available at: http://scikit-learn.org/stable/modules/svm.html

Silvio, Giuliano & Giuliano, Grignaschi (2016). *Animal testing is the best way to find new treatments for patients*. [Cited 12.01.2018]. Available at: https://ac-els-cdn-com.proxy.uwasa.fi/S0953620516304101/1-s2.0-S0953620516304101-main.pdf?_tid=1838c604-04f1-11e8-8d72-00000aacb35e&acdnat=1517229598_f768065703bd177de20a83248cb349b5

Statsoft (n.d.). *Support Vector Machines (SVM).*[Cited 03.03.2018]. Available at: http://www.statsoft.com/Textbook/Support-Vector-Machines

Sunil, Kumar (2018). *What is an example application of unsupervised machine learning?* [Cited 01.04.2018]. Available at: https://www.quora.com/What-is-an-example-application-of-unsupervised-machine-learning

Sunil, Ray (2017). *Essentials of machine learning algorithms (with Python and R codes)*. [Cited 05.02.2018]. Available at: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

ThermoFisher (n.d.). *Introduction to cell culture*. [Cited 29.12.2017]. Available at: https://www.thermofisher.com/us/en/home/references/gibco-cell-culture-basics/introduction-to-cell-culture.html

Tmedweb,(2013).*Hepatic Drug Clearance*.[Cited 11.04.2018]. Available at: http://tmedweb.tulane.edu/pharmwiki/doku.php/hepatic_drug_clearance

Trivedi, Jay (2011). *Data acquisition system and data logger*. [Cited 13.03.2018]. Available at: https://www.slideshare.net/jay022/data-acquisition-system-data-logger

Tripod(n.d.). *Support Vector Machine Regression.* [Cited 25.03.2018]. Available at: http://kernelsvm.tripod.com/

Wei ,S. Cheng, Y. Guo, A. Min, Z. Hong, Z. Yue, W & Ping, H (2016*). Organs-on-chips and its application*. China: Chinese Journal of Analytical Chemistry. [Cited 30.12.2017]. Available at: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S1872204016609209

Wyss Institute, (2017). *Living, breathing human lung-on-a-chip: A potential drug-testing alternative*. [Cited 27.12. 2017]. Available at: https://wyss.harvard.edu/living-breathing-human-lung-on-a-chip-a-potential-drug-testing-alternative/

Ye, Fang. Richard, M & Jean, Klein (2017). *Three- dimensional cell cultures in drug discovery and development*. USA: Sage. [Cited 02.01.2018]. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5448717/

Yin, Yu. Kazim, Moncal. Jianqiang, Li. Weijie, P. Iris, Rivero. James, Martin & Ibrahim, Ozbolat (2016). *Three-dimensional bioprinting using self assembling scalable scaffold-free "tissue strands" as a new bioink*. [Cited 10.01.2018]. Available at: https://www.nature.com/articles/srep28714

Appendix

```
data = read.table(file = file.choose(),header = F, sep = " ", na.strings = "", stringsAsFactors = F)
# adding columns and assigning them
data$age = data[10001:20000,2]
data$p1 = data[20001:30000,2]
data$p2 = data[30001:40000,2]
data$p3 =data[40001:50000,2]
data$p4 =data[50001:60000,2]
data$p5 =data[60001:70000,2]
data$p6 =data[70001:80000,2]
data$p7 =data[80001:90000,2]
data$Output =data[90001:100000,2]
# Remove the overlapped colums
my_data = data[-c(10001:100000), ]
# Delete the first column
my_data$V1 <- NULL
# Giving names to the columns
naming = c('Gender', 'Age','P1','P2','P3','P4','P5','P6','P7','Output')
colnames(my_data) = naming
my_data
any(is.na(my_data))


##convert to categorical

my_data$Gender <- factor(my_data$Gender)
my_data$Output <- factor(my_data$Output)
str(my_data)

# Visualizing the data

library(ggplot2)
library(digest)
library(ggplot2)
library(ggthemes)
library(dplyr)
library(caTools)

## Solubility between male and female
pl<- ggplot(my_data, aes(P1))
pl<- pl + geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
pl<- pl +theme_bw()
pl
```

```
## Drug partitioning into organs

b<- ggplot(my_data, aes(P2))
b<- b+ geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
b<- b +theme_bw()
b

## Iron level in both men and women


l<- ggplot(my_data, aes(P3))
l<- l+ geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
l<- l +theme_bw()
l

## zinc Level in the blood stream
z<- ggplot(my_data, aes(P4))
z<- z+ geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
z<- z +theme_bw()
z

## Magnesium Level in the blood stream
m<- ggplot(my_data, aes(P5))
m<- m+ geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
m<- m +theme_bw()
m

## Unbound fraction of drug

u<- ggplot(my_data, aes(P6))
u<- u + geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
u<- u +theme_bw()
u

## Extraction Ratio

e<- ggplot(my_data, aes(P7))
e<- e+ geom_histogram(aes(fill = Gender),color ='black', bins =20, alpha = 0.5)
e<- e +theme_bw()
e
#  Output analysis
PO <- ggplot(my_data, aes(x = factor(Output)))
PO <- PO + geom_bar(aes(fill = Gender), position = 'dodge')
PO <- PO + theme_bw()
PO
```

```
##  Scatered plot of different parameters against the drug solubility
# Age difference against solubility
AS = ggplot(my_data,aes(Age, P1)) + geom_point(aes(color = Output)) + theme_bw()
AS
# Drug partitioning into different body organs against drug solubilty
PS = ggplot(my_data,aes(P2, P1)) + geom_point(aes(color = Output)) + theme_bw()
PS
# Iron level against drug solubilty
IS = ggplot(my_data,aes(P3, P1)) + geom_point(aes(color = Output)) + theme_bw()
IS
# Zinc level against drug solubilty
ZS = ggplot(my_data,aes(P4, P1)) + geom_point(aes(color = Output)) + theme_bw()
ZS
# Magnesium level against drug solubilty
MS = ggplot(my_data,aes(P5, P1)) + geom_point(aes(color = Output)) + theme_bw()
MS
# Unbound fraction of drug against drug solubilty
US = ggplot(my_data,aes(P6, P1)) + geom_point(aes(color = Output)) + theme_bw()
US
# Extraction ratio against drug solubilty
ES = ggplot(my_data,aes(P7, P1)) + geom_point(aes(color = Output)) + theme_bw()
ES
## Building the SVM Model
# splitting of test and training data
#set a seed

set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.9)
train <- subset(my_data, sample == TRUE)
test <- subset(my_data, sample == FALSE)
#Training the model using support vector machine
library(e1071)
model <- svm(Output ~ ., data = train)
print(summary(model))

# Testing the model

predict_values <- predict(model,test[1:9])
# Cofusion Matrix
table(predict_values,test$Output)
# comparable table
results <- cbind(predict_values, test$Output)
colnames(results)<- c('predicted', 'actual')
results <- as.data.frame(results)
print(head(results))
```

```
#Tuning the model
tuned.result <- tune(svm, train.x = Output ~ ., data = train, kernel = 'radial', ranges = list(cost=
c(0.5,1,1.5), gamma = c(0.1,0.5, 0.7)))
print(summary(tuned.result))
#Train the tuned parameter
tuned_model <- svm(Output ~ ., data = train, cost = 1.5, gamma = 0.1)
tuned_predictions <- predict(tuned_model,test[1:9])
# Confusion matrix
table(tuned_predictions, test$Output)
# Change the gamma parameter
tuned_model1 <- svm(Output ~ ., data = train, cost = 1.5, gamma = 0.5)
tuned_predictions1 <- predict(tuned_model1,test[1:9])
# Confusion matrix
table(tuned_predictions1, test$Output)

## Accuracy of of the output before tunning
# Accuracy =  (correctly classified/ total tested sample) *100
Model_Accuracy1 = (952/1000) *100
Model_Accuracy1
## Accuracy of of the output with cost = 1.5
# Accuracy =  (correctly classified/ total tested sample) *100
Model_Accuracy2 = (963/1000) *100
Model_Accuracy2

## Accuracy of  the output with gamma = 0.4
# Accuracy =  (correctly classified/ total tested sample) *100
Model_Accuracy3 = (915/1000) *100
Model_Accuracy3
## Sensitivity  and specificy of the respective classes from the best model
# dead cell (class 0)
Tp_0 = 87
Tn_0 = 876
Fp_0 =  4
Fn_0 = 3
Sentivity_0 = (Tp_0/(Tp_0 + Fn_0)) * 100
Specificy_0 = (Tn_0/(Fp_0 + Tn_0)) * 100
Sentivity_0
Specificy_0

# Cell unaffected (class 1)

Tp_1 = 61
Tn_1 = 902
Fp_1 = 13
Fn_1 = 8
Sentivity_1 = (Tp_1/(Tp_1 + Fn_1)) * 100
```

```
Specificy_1 = (Tn_1/(Fp_1 + Tn_1)) * 100
Sentivity_1
Specificy_1

# Negative  effect on the cell (class 2)
Tp_2 = 293
Tn_2 = 670
Fp_2 = 11
Fn_2 = 19
Sentivity_2 = (Tp_2/(Tp_2 + Fn_2)) * 100
Specificy_2 = (Tn_2/(Fp_2 + Tn_2)) * 100
Sentivity_2
Specificy_2

# Positive  effect on the cell (class 2)
Tp_3 = 522
Tn_3 = 441
Fp_3 = 9
Fn_3 = 7
Sentivity_3 = (Tp_3/(Tp_3 + Fn_3)) * 100
Specificy_3 = (Tn_3/(Fp_3 + Tn_3)) * 100
Sentivity_3
Specificy_3
# Classification for the model
# With C = 1.5 and gamma = 0.1
classification_error1 <- 1- sum(tuned_predictions == test$Output)/length(tuned_predictions)
classification_error1
# Using a training sample of 5%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.05)
train5 <- subset(my_data, sample == TRUE)
test5 <- subset(my_data, sample == FALSE)
train_model5 <- svm(Output ~ ., data = train5, cost = 1.5, gamma = 0.1)
train5_predictions <- predict(train_model5,test5[1:9])
# Confusion matrix
table(train5_predictions, test5$Output)
# Classifion error for 5% train data
classification_error5 <- 1- sum(train5_predictions == test5$Output)/length(train5_predictions)
classification_error5

# Using a training sample of 10%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.10)
train10 <- subset(my_data, sample == TRUE)
test10 <- subset(my_data, sample == FALSE)
train_model10 <- svm(Output ~ ., data = train10, cost = 1.5, gamma = 0.1)
```

```
train10_predictions <- predict(train_model10,test10[1:9])
# Confusion matrix
table(train10_predictions, test10$Output)
# Classifion error for 10% train data
classification_error10 <- 1- sum(train10_predictions ==
test10$Output)/length(train10_predictions)
classification_error10

# Using a training sample of 20%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.20)
train20 <- subset(my_data, sample == TRUE)
test20 <- subset(my_data, sample == FALSE)
train_model20 <- svm(Output ~ ., data = train20, cost = 1.5, gamma = 0.1)
train20_predictions <- predict(train_model20,test20[1:9])
# Confusion matrix
table(train20_predictions, test20$Output)
# Classifion error for 20% train data
classification_error20 <- 1- sum(train20_predictions ==
test20$Output)/length(train20_predictions)
classification_error20

# Using a training sample of 30%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.30)
train30 <- subset(my_data, sample == TRUE)
test30 <- subset(my_data, sample == FALSE)
train_model30 <- svm(Output ~ ., data = train30, cost = 1.5, gamma = 0.1)
train30_predictions <- predict(train_model30,test30[1:9])
# Confusion matrix
table(train30_predictions, test30$Output)
# Classifion error for 30% train data
classification_error30 <- 1- sum(train30_predictions ==
test30$Output)/length(train30_predictions)
classification_error30

# Using a training sample of 40%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.40)
train40 <- subset(my_data, sample == TRUE)
test40 <- subset(my_data, sample == FALSE)
train_model40 <- svm(Output ~ ., data = train40, cost = 1.5, gamma = 0.1)
train40_predictions <- predict(train_model40,test40[1:9])
# Confusion matrix
table(train40_predictions, test40$Output)
# Classifion error for 50% train data
```

```
classification_error40 <- 1- sum(train40_predictions ==
test40$Output)/length(train40_predictions)
classification_error40

# Using a training sample of 50%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.50)
train50 <- subset(my_data, sample == TRUE)
test50 <- subset(my_data, sample == FALSE)
train_model50 <- svm(Output ~ ., data = train50, cost = 1.5, gamma = 0.1)
train50_predictions <- predict(train_model50,test50[1:9])
# Confusion matrix
table(train50_predictions, test50$Output)
# Classifion error for 50% train data
classification_error50 <- 1- sum(train50_predictions ==
test50$Output)/length(train50_predictions)
classification_error50

# Using a training sample of 60%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.60)
train60 <- subset(my_data, sample == TRUE)
test60 <- subset(my_data, sample == FALSE)
train_model60 <- svm(Output ~ ., data = train60, cost = 1.5, gamma = 0.1)
train60_predictions <- predict(train_model60,test60[1:9])
# Confusion matrix
table(train60_predictions, test60$Output)
# Classifion error for 60% train data
classification_error60 <- 1- sum(train60_predictions ==
test60$Output)/length(train60_predictions)
classification_error60
# Using a training sample of 70%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.70)
train70 <- subset(my_data, sample == TRUE)
test70 <- subset(my_data, sample == FALSE)
train_model70 <- svm(Output ~ ., data = train70, cost = 1.5, gamma = 0.1)
train70_predictions <- predict(train_model70,test70[1:9])
# Confusion matrix
table(train70_predictions, test70$Output)
# Classifion error for 70% train data
classification_error70 <- 1- sum(train70_predictions ==
test70$Output)/length(train70_predictions)
classification_error70

# Using a training sample of 80%
```

```
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.80)
train80 <- subset(my_data, sample == TRUE)
test80 <- subset(my_data, sample == FALSE)
train_model80 <- svm(Output ~ ., data = train80, cost = 1.5, gamma = 0.1)
train80_predictions <- predict(train_model80,test80[1:9])
# Confusion matrix
table(train80_predictions, test80$Output)
# Classifion error for 80% train data
classification_error80 <- 1- sum(train80_predictions ==
test80$Output)/length(train80_predictions)
classification_error80
# Using a training sample of 90%
set.seed(101)
sample <- sample.split(my_data$Output,SplitRatio = 0.90)
train90 <- subset(my_data, sample == TRUE)
test90 <- subset(my_data, sample == FALSE)
train_model90 <- svm(Output ~ ., data = train90, cost = 1.5, gamma = 0.1)
train90_predictions <- predict(train_model90,test90[1:9])
# Confusion matrix
table(train90_predictions, test90$Output)
# Classifion error for 90% train data
classification_error90 <- 1- sum(train90_predictions ==
test90$Output)/length(train90_predictions)
classification_error90
train_percentage <-  c(5, 10,20,30,40,50,60,70,80,90)
classification_error <- c(classification_error5, classification_error10, classification_error20,
classification_error30, classification_error40,classification_error50, classification_error60,
classification_error70, classification_error80, classification_error90)
plot(train_percentage, classification_error, 'l')
```