

TRINITY COLLEGE DUBLIN



**Diagnosis And Mortality Prediction
By Applying Machine Learning
On a Hepatocellular Carcinoma
And a Liver Disease Datasets**

Tomasz Wisniowski

BSc (Hons) Computer Science
Final Year Project April 2020
Supervisor: Dr. Lucy Hederman

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at <http://www.tcd.ie/calendar>.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at <http://tcd-ie.libguides.com/plagiarism/ready-steady-write>.



Signed

27 / 04 / 2020

Date

Abstract

Liver diseases and liver cancer continue to increase their death toll world wide every year. One of these malignancies, Hepatocellular Carcinoma (HCC), has a death rate of 87.5%. This primary liver cancer is so deadly due to the lack of symptoms and an optimal diagnostic technique. Newly developed or improved technologies, such as machine learning could provide a more consistent and error free solutions for patient diagnosis and in turn increase the survival rate of these patients.

The objective of this study was to develop new machine learning models for:

- (a) Diagnosis of liver disease.
- (b) Prediction of HCC patient survival.

The datasets used to build the models were obtained from two free source repositories, Kaggle and UCI - Machine learning Repository. The Indian Liver Patients datasets contained 416 liver patient records and 167 non liver patient records and also posses a certain gender imbalance as it contained 441 male patient records and 142 female patient records. The HCC survival dataset contained real clinical data of 165 patients diagnosed with HCC while.

Both of the datasets were balanced using the Synthetic Minority Over-sampling Technique (SMOTE). Different techniques for missing values were used.

Finding the most frequent value with regard to each target variable proved to give the most accurate results.

Five popular machine learning algorithms were chosen for the classification task: Logistic Regression, Support Vector Machines, Multi layered Perceptron, K-Nearest Neighbors, Random Forest. Their optimal hyper-paramaters were chosen using the grid search method.

The model for Liver Disease Diagnosis was the most accurate when Support Vector Machine algorithm was applied. This model achieved the F1 score of 83.38%(+/- 0.01%) and F2 score 93.10% while the model for HCC Survival Prediction performed best when k-Nearest Neighbour ($k = 1$) was used.

This model achieved F1 score of 85.899% (+/- 0.11%) and F2 score of 94.33%.

The results and parameter optimization were verified multiple times using 5-Fold Cross-Validation.

Acknowledgements

Firstly, I would like to thank my supervisor Dr. Lucy Hederman who continuously overlooked my work while also providing me with new ideas and more insight into machine learning and general research. I would also like to thank Dr. Inmaculada Arnedillo-Sanchez, who took her time to view my presentation and highlighted a crucial area which needed further investigation.

Secondly, I would like to thank the IBM Innovation Exchange (IIX) team, who not only sparked the idea for this project but also provided constant support and shared their expertise on which machine learning techniques could potentially result in success and which would not, saving me a lot of frustration.

I am also forever grateful to my peers in Computer Science, who throughout the years supported me many times. Without their friendship, my time in Trinity College would not have been as enjoyable as it has been. I wish you all the best of luck in all your future endeavors.

Lastly, I would like to thank my parents for their never ending love and support. Words can not capture how grateful I am to have them.

Contents

Declaration	I
Abstract	II
Acknowledgements	III
List of Figures	VI
List of Tables	VII
1 Introduction	1
1.1 Research Motivation	1
1.2 Research Question	4
1.3 Research Objectives	4
1.4 Methodology	5
1.5 Overview of Report	7
2 Literature Review	9
2.1 Artificial Intelligence and Machine Learning	9
2.2 State of the Art	11
2.2.1 A Model Which Predicts Cancer Susceptibility	13
2.2.2 A Model Which Predicts Cancer Prognosis	13
2.2.3 A Model Which Predicts Cancer Survival rates	15
2.2.4 A Model Which Predicts Cancer Recurrence	16
2.3 Study Review: Diagnosis of Hepatocellular Carcinoma	17
2.4 Study Review:	
Hepatocellular Carcinoma Survival Prediction	19
2.5 Description of Algorithms	21
2.5.1 Logistic Regression	21
2.5.2 Support Vector Machine	22
2.5.3 Random Forest	23
2.5.4 Multi-layer Perceptron	24
2.5.5 K-Nearest Neighbors	25
2.6 Common Machine Learning Challenges	26
2.6.1 Lack Of Data	26
2.6.2 Bias vs. Variance	27
2.6.3 The Curse of Dimensionality	28
2.6.4 Overfitting vs. Underfitting	28
2.6.5 Computational Complexity	29
2.7 Summary	29

3	Design and Implementation	30
3.1	Technology Selection	30
3.2	Data Collection	32
3.2.1	HCC Survival Dataset	32
3.2.2	Liver Disease Dataset	32
3.2.3	Handling The Unbalanced Datasets	32
3.3	Data Pre-Processing	34
3.3.1	Data Cleaning	34
3.3.1.1	Missing Data	34
3.3.2	Feature Engineering	36
3.3.2.1	Feature Encoding	36
3.3.2.2	Feature Scaling	37
3.3.3	Feature Selection	40
3.4	Model Training	44
3.4.1	Hyperparameter Tuning	44
3.4.2	Cross-Validation	45
3.5	Summary	46
4	Evaluation	46
4.1	Evaluation Metrics	46
4.1.1	Predictive accuracy	47
4.1.2	Recall	47
4.1.3	Precision	48
4.1.4	F Score	48
4.2	Algorithm Comparison	49
4.3	Importance of Data Pre-Processing Techniques	52
4.4	Importance of Training/Testing Techniques	53
5	Conclusion	55
5.1	Future Work	57
5.2	Final Remarks	59
	Bibliography	61
	Appendices	66
A	Feature Correlation heat maps, used for feature selection.	66
B	Predictive Accuracy of Models Built.	68
C	Feature Importance for Liver Disease Dataset.	69

List of Figures

1.1	Survival rates of various cancers	2
1.2	Steps taken in the Implementation Process.	6
2.1	AI vs ML vs DL.	9
2.2	Example of Artificial Neural Networks.	13
2.3	Type of Cancer vs Amount of Research Papers and Algorithms Used. ⁵ . .	14
2.4	Support vectors, hyperplane and margins.	22
2.5	Kernelling or “The Kernel Trick”.	22
2.6	Representation of Random Forest.	23
2.7	Multilayer-Perceptron-Network.	24
2.8	K-Nearest Neighbors.	25
2.9	Bias vs. Variance.	27
2.10	Overfitting vs. Underfitting.	28
3.1	Addressing class imbalance via the SMOTE method.	33
3.2	Flawed Label Encoding method.	36
3.3	One Hot Encoding used on the ‘performance status’ feature.	37
3.4	Features before Standardization.	39
3.5	Features after Standardization.	39
3.6	Accuracy to number of features selected for various algorithms (HCC Survival)	41
3.7	Accuracy to number of features selected for various algorithms (Liver Disease):	42
3.8	Ranking the importance of different feature visualized (HCC Survival). .	43
4.1	F1 Score obtained by all algorithms after 5-fold cross validation (HCC Survival).	49
4.2	F1 Score obtained by all algorithms after 5-fold cross validation (Liver Disease).	50
4.3	F2 Score obtained by all algorithms after 5-fold cross validation (HCC Survival).	51
4.4	F2 Score obtained by all algorithms after 5-fold cross validation (Liver Disease).	51
4.5	Impact of data pre-processing on the F1 score of various algorithms (HCC Survival).	52
4.6	5-Fold Cross Validation performed on the HCC Survival Dataset.	53
4.7	5-Fold Cross Validation performed on the Liver Disease Dataset.	54

List of Tables

2.1	Definition of Intelligence with regard to AI. (Russell and Norvig, 2009)	10
2.2	Machine Learning Methods used for cancer prediction.	12
2.3	Results when 49 features were used.	20
2.4	Results when 7 features were used.	20
3.1	Percentage of missing values in the HCC Survival dataset.	35
3.2	Predictive accuracy increase achieved through Hyperparameter Tuning. (HCC Survival)	45
4.1	Types of classification made by models.	46

Chapter 1

Introduction

This project explores how machine learning can be used to:

- (a) diagnose if a patient is afflicted with liver disease
- (b) predict the survival of patients afflicted with Hepatocellular Carcinoma based on a number of various features from two different datasets.

1.1 Research Motivation

Cancer begins when healthy cells mutate and grow out of control.

A healthy cell does not turn into a cancer cell overnight. Its behaviour gradually changes, which is a direct result of damage to between three and seven of the hundreds of genes that control cell growth, division and life span.

First, the cell starts to grow and multiply. Over time, more changes may take place. The cell and its descendants may eventually become immortal, escape destruction by the body's defence and develop their own blood supply. These cells form a mass called a tumor. A tumor can be cancerous or benign. A cancerous tumor is malignant, meaning it can grow and spread to other parts of the body. A benign tumor can grow but will not spread to other organs.¹

Hepatocellular Carcinoma (from here on also referred to as HCC) is the most common type of primary liver cancer, meaning it began in the liver itself. It accounts for 80% to 90% of primary liver malignancies.²

Although it is only the 15th most common cancer it is the 4th deadliest cancer in the world with a survival rate of only 12.1%.

¹Science Museum, n.d. How Do Healthy Cells Become Cancerous?. Available at: <http://whoami.sciencemuseum.org.uk/whoami/findoutmore/yourbody/whatisacancer/whathappensincancer/howdohealthycellsbecomecancerous>.

²Gu, J., 2013. Primary Liver Cancer. Dordrecht: Springer, pp.399-400.

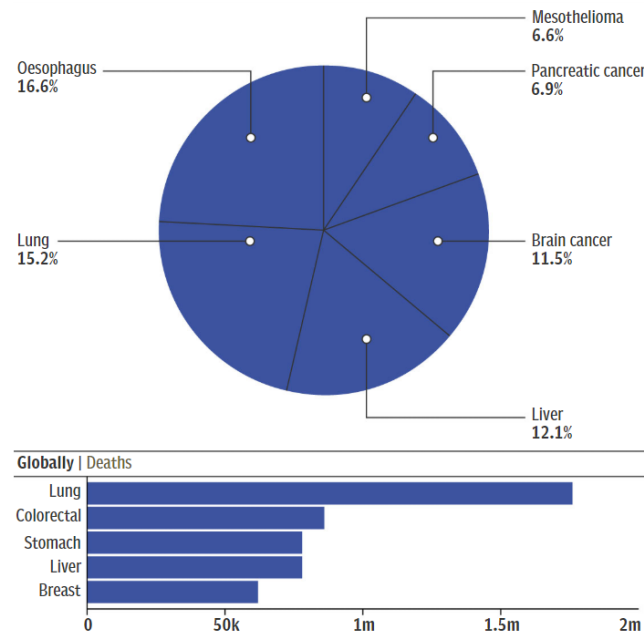


Figure 1.1: Survival rates of various cancers³

It is estimated that every year, 800,000 people worldwide are afflicted with HCC. From these only around 100,000 will survive. Incidence rates vary geographically: the highest occur in Southeast Asia, China, Sub-Saharan Africa, and are much lower in Europe.

HCC occurs most often in people with chronic liver diseases. HCC usually develops on the basis of post-inflammatory cirrhosis, whose most important etiological factors are hepatitis B and C. On top of that it is the most common cause of death in people with cirrhosis. Both hepatitis B and hepatitis C viruses are responsible for nearly 80% of cases of the disease. Alcoholic cirrhosis and non-alcoholic fatty liver also play an important role in Western countries.

Initially, the cancer develops scarcely, and its symptoms are difficult to distinguish from the usual accompanying liver cirrhosis.

The most important clinical symptoms include deterioration of the general condition of the patient with cirrhosis, pain in the right hypochondrium, symptoms resulting from cholestasis, jaundice, hepatomegaly and ascites. (Pinter M, Trauner M, et al. 2016)

³Mintz, L., 2020. World Cancer Day: What Is The Most Common Cancer In The UK, And Which Has The Worst Survival Rate?. The Telegraph. Available at: <https://www.telegraph.co.uk/health-fitness/body/world-cancer-day-common-cancer-uk-has-worst-survival-rate/>.

However, if certain conditions are met and its characteristic features are found, the cancer can be diagnosed on the basis of a radiographic image by computed tomography or magnetic resonance imaging. In other cases, a biopsy must be performed subjected to cytological examination.

All patients with cirrhosis should be subjected to ultrasound screening every 6 months. While the “ α -fetoprotein test” is no longer recommended due to the high percentage of false positives delivered.

While there are quite a few methods for diagnosis of HCC, all of them take valuable time. Although historically a biopsy of the tumor is required to prove the diagnosis, imaging (especially MRI) findings may be conclusive enough to obviate histopathologic confirmation.⁴

Naturally, a faster more accurate method of diagnosis is needed.

A machine learning model with a high degree of accuracy and high F scores could decrease the the time needed for diagnosis or even fully cut the need for imaging. This would allow patients to get a biopsy right away or potentially begin their treatment early, increasing their overall chances of survival.

The second step would include training another model, which would predict patients’ mortality rate. The results would give doctors a better idea of the direction a patients’ health is heading in. It would also allow doctors to make better decisions when prioritizing patients for liver transplants. Finally the last step would include creating a model which would deal with biomarker prediction. When a liver transplant isn’t possible right away, the model could theoretically be used to manipulate biomarkers thus increasing patients’ survivability.

⁴Mayoclinic.org. n.d. Hepatocellular Carcinoma - Overview - Mayo Clinic. [online] Available at: <https://www.mayoclinic.org/diseases-conditions/hepatocellular-carcinoma/cdc-20354552>.

1.2 Research Question

“To what extent can machine learning techniques be used to predict liver disease and Hepatocellular Carcinoma patients’ survivability?”

1.3 Research Objectives

The primary objective was to verify whether machine learning (ML) can be used as a viable option for the prediction of liver disease and the prediction of patients’ survivability. However, the main goal had to be divided into multiple smaller task as each step of a machine learning pipeline is equally important.

1. **A thorough examination of previous literature:** Research into previous findings should always be conducted prior to a study to ensure that actions and previous mistakes aren’t repeated. Success of other approaches should always be considered. When investigating every approach questions such as: “Why was this study successful?”, “Why was it unsuccessful?”, “What could be improved?” should be asked.
2. **Data collection:** Medical datasets and databases are quite scarce due to the fact that hospitals and doctors can not provide open access to it. Patients do not want to share their medical history for fear of losing their privacy. This can be a huge challenge for researchers and all data scientists. Various open source datasets had to be inspected and analysed in order to get the best, most accurate results.
3. **Use of different techniques for:**
 - (a) Cleaning the data.
 - (b) Missing Data.
 - (c) Data Formatting.
 - (d) Outlier detection.
 - (e) Feature Engineering.
 - (f) Feature Encoding.
 - (g) Feature Scaling.
 - (h) Feature selection.
 - (i) Hyper-parameter Tuning.
 - (j) Cross-Validation.
 - (k) Algorithm selection and comparison.

1.4 Methodology

To make the study applicable, the research had to be performed using the standard practices applied in machine learning.

Like any machine learning model, the research began by acquiring data. As said before medical datasets are relatively scarce due to the fact that they may infringe on privacy. Nonetheless clean relevant data had to be gathered. The data for liver disease prediction and the data for HCC survival was not presented in one dataset. The two datasets could not be combined or appended as both described two different sub-groups of people. One described a group of patients from India with or without liver disease, while the other described a group of patients from Portugal where all had HCC. The two datasets were acquired in the form of CSV datasets via Kaggle and UC Irvine Machine Learning Repository, online data science and ML communities.

The “Indian Patients Dataset” contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India while the “HCC Survival Dataset” contains 165 real patient records and 40 synthetic patients generated using oversampling method called SMOTE (which will be explained later), collected at a university hospital in Portugal.

The next step included data preparation. This proved to be the key step of the research. When working with such a small amount of data, every record has to be used to maximise results, this means that steps such as “Missing Data Replacement” or “Outlier Detection/Mining ” become crucial.

Just like Tamraparni Dasu and Theodore Johnson said in their book, “Exploratory Data Mining and Data Cleaning” released in 2003, data preparation should take up 80% of the total time spent on the project.

The third step involved investigating the algorithms that may be optimal for these machine learning models. Literature and previous research was thoroughly investigated to discover which algorithms have been applied and succeeded before.

The fourth step focused on training and model comparison.

Each algorithm had to be trained separately to ensure that underfitting and overfitting does not occur. Having multiple train/test data splits such as 60/40, 70/30, 80/20 and cross-validation corrected this problem.

Hyper-parameter-tuning focuses on calibrating each individual parameter for optimal results in regard to both accuracy and F scores on the given datasets.

The final step concentrated on model evaluation. This helped in verifying whether a machine learning approach is a applicable method of HCC prediction and survivability.

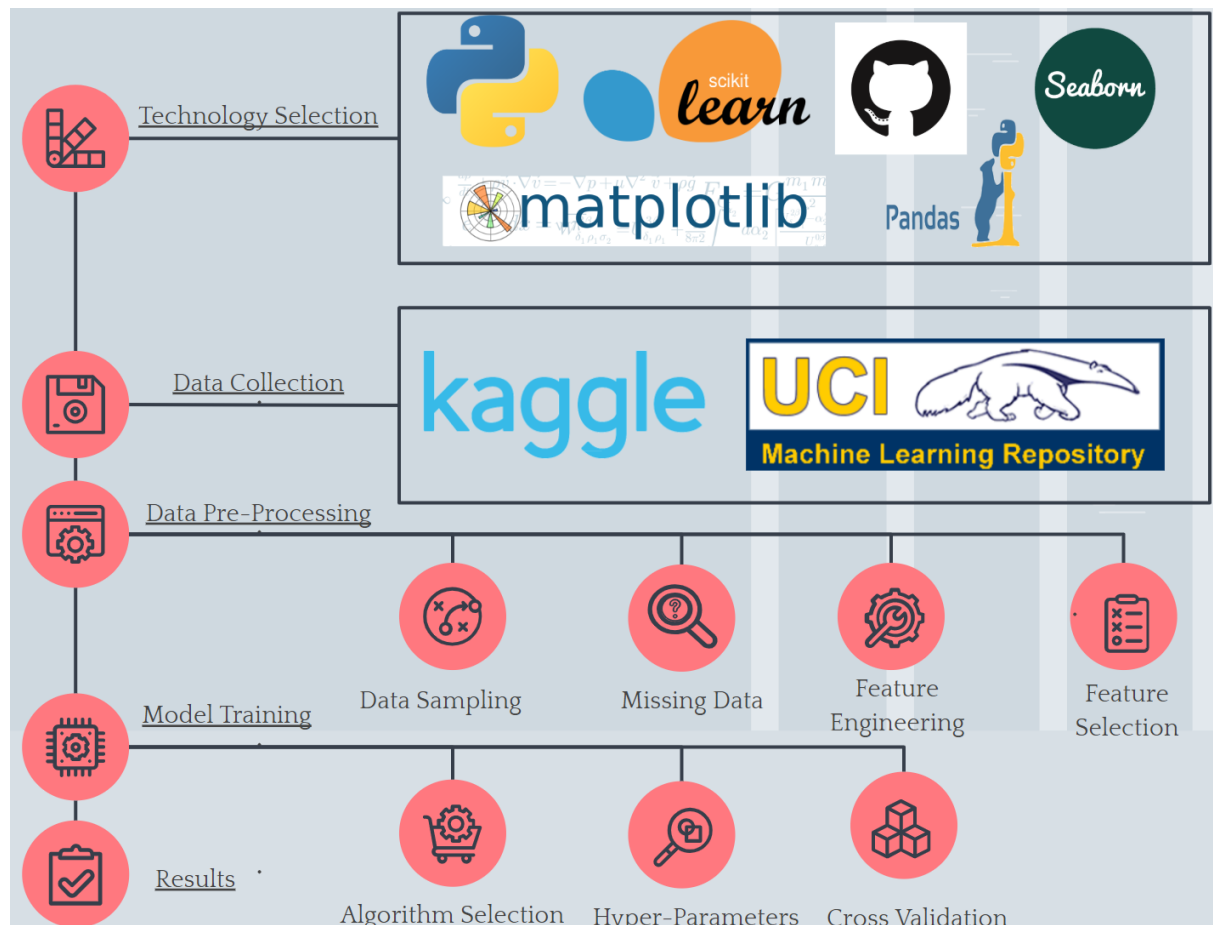


Figure 1.2: Steps taken in the Implementation Process.

1.5 Overview of Report

This report presents a complete account of all the stages of the research undertaken, the process of implementation, reasoning behind decisions made and finally, evaluation of the results.

Chapter 2: Here the report deals with an extensive look into previous literature. A brief review of the history of artificial intelligence and machine learning provides a good introduction to the report.

Illustrating multiple examples where machine learning has been applied gives even greater insight into the potential of this technology.

Finally, a review of previous research and approaches used in the area of HCC prediction and HCC survival is conducted. The results obtained from these studies are comprehensively examined and evaluated.

Multiple challenges are also identified. Only after doing so the process of implementation could begin.

Chapter 3: In this chapter the report describes the design choices made and the process of implementing those choices. Firstly the selection of suitable technologies such as the programming language, libraries and hardware had to be performed.

Secondly, the chapter includes a description of the process of locating and collecting relevant medical data.

Thirdly comes the selection of optimal techniques for the different stages outlined in “Research Objectives” under the “Use of different techniques for:” section on page 10. Finally a detailed description of the training and testing process for each algorithm selected.

Chapter 4: This chapter consists of thorough evaluation of the work done in the previous chapter. The performance of the algorithms being compared is measured using several metrics, with F2 score used as the key metric. The evaluation of the different data pre-processing and training/testing techniques implemented is also performed in this chapter along with estimating their individual contribution to the performance of the algorithms.

Chapter 5: The last chapter is used to reflect and assess the level of achievement of the research objectives determined in the “Research Objectives” section on page 10. Areas where further work could be conducted are also identified. Although some improvements could not have been implemented within the constraints of this project and the data collected. The research could be further improved if more data was accumulated and there was less of a time constraint. A final personal reflection upon the process and challenges concludes the project.

Chapter 2

Literature Review

2.1 Artificial Intelligence and Machine Learning

The terms Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning are sometimes used interchangeably. However, artificial intelligence refers to any program or application with the ability to think, learn and act rationally like a human (Russell and Norvig, 2009) whilst ML is best described by Tom Mitchell in “Machine Learning” (1997):

“A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T , as measured by P , improves with experience E .”

Where experience E is the data used to perform task T . From these two definitions a clear conclusion can be made about the fact that machine learning is just a subset of artificial intelligence. Moreover, deep learning can be described as a subset of machine learning where artificial neural networks adapt and learn from vast amounts of data.

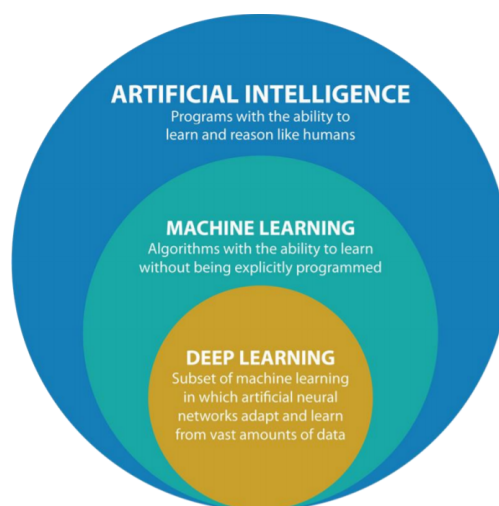


Figure 2.1: AI vs ML vs DL.
(Beel J. Trinity College Dublin - Blackboard Notes)

Beneath we have several definitions of Artificial Intelligence made by individuals who made outstanding achievements in the area of AI and development of thinking computer systems:

Table 2.1: Definition of Intelligence with regard to AI. (Russell and Norvig, 2009)

Systems that think like humans	Systems that think rationally
<p>“The exciting new effort to make computers think . . . <i>machines with minds</i>, in the full and literal sense.” (Haugeland, 1985)</p> <p>“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . .” (Bellman, 1978)</p>	<p>“The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985)</p> <p>“The study of the computations that make it possible to perceive, reason, and act.” (Winston, 1992)</p>
Systems that act like humans	Systems that act rationally
<p>“The art of creating machines that perform functions that require intelligence when performed by people.” (Kurzweil, 1990)</p> <p>“The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)</p>	<p>“Computational Intelligence is the study of the design of intelligent agents.” (Poole <i>et al.</i>, 1998)</p> <p>“AI . . . is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)</p>

Due to the fact that this research concerns machine learning, it is appropriate to explain it in more detail. Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions.

Machine learning can then be split into three different types of learning:

1. **Supervised Learning** - model learns from being trained on labelled training data. Here the researcher acts as a guide to teach the algorithm what conclusions or predictions it should come up with.

(Support vector machine, Linear and logistics regression, Neural network, Classification trees and random forest etc.)

2. **Unsupervised Learning** - model finds similarities in the data and groups/clusters them together. Here there is no teacher, algorithms are left to their own to discover and present the hidden structure in the data. (Cluster algorithms, K-means, Hierarchical clustering, Dimensionally reduction algorithms, Anomaly detections, etc.)

3. **Reinforcement Learning** - model learns by receiving feedback in the form of reward and punishment. This happens by taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path the model should take in a specific situation.

2.2 State of the Art

Machine learning has already been introduced to the world of pathology. It quickly became clear that ML has some advantage over pathologists and could be used to improve diagnostic consistency and reduce errors.

The real drivers for this include:

- (i) An acute shortage of pathologists in many countries.
- (ii) Ageing populations driving up pathology workloads.
- (iii) Increased cancer screening programs resulting in increased workloads.
- (iv) Rising complexity of pathology tests increase the time taken per case.
- (v) The need for pathology laboratories to outsource expertise.

Testing the sample acquired from a biopsy and then writing a report may take a pathologist even up to 10 days. This is because different types of body tissues take longer to process than others. A computer can perform thousands of biopsies reviews in a matter of seconds.

Secondly, there is so much data out in the world that humans can't possibly go through it all. Machines can do so while simultaneously performing accurate computations and detecting patterns in data.

There already exist many models which can accurately predict cancer susceptibility, prognosis, survival and recurrence. Some of these models, which achieved very high accuracy, are presented next, with models regarding cancer prediction and survival being described in greater detail as they relate more to this research. Many other models could be mentioned however the selected few give a reasonable insight into what machine learning can do.

Please refer to the table provided on the next page to examine how many times machine learning has been used to perform predictions or classification on various cancer related datasets.

Please also notice how Artificial Neural Networks are often the go to algorithm, regularly providing an overall accuracy increase.

Table 2.2: Machine Learning Methods used for cancer prediction.⁵

Table 2: Survey of machine learning methods used in cancer prediction showing the types of cancer, clinical endpoints, choice of algorithm, performance and type of training data.

Cancer Type	Clinical Endpoint	Machine Learning Algorithm	Benchmark	Improvement (%)	Training Data	Reference
bladder	recurrence	fuzzy logic	statistics	16	mixed	Catto et al, 2003
bladder	recurrence	ANN	N/A	N/A	clinical	Fujikawa et al, 2003
bladder	survivability	ANN	N/A	N/A	clinical	Ji et al, 2003
bladder	recurrence	ANN	N/A	N/A	clinical	Spyridonos et al, 2002
brain	survivability	ANN	statistics	N/A	genomic	Wei et al, 2004
breast	recurrence	clustering	statistics	N/A	mixed	Dai et al, 2005
breast	survivability	decision tree	statistics	4	clinical	Delen et al, 2005
breast	susceptibility	SVM	random	19	genomic	Listgarten et al, 2004
breast	recurrence	ANN	N/A	N/A	clinical	Mattfeldt et al, 2004
breast	recurrence	ANN	N/A	N/A	mixed	Ripley et al, 2004
breast	recurrence	ANN	statistics	1	clinical	Jerez-Aragones et al, 2003
breast	survivability	ANN	statistics	N/A	clinical	Lisboa et al, 2003
breast	treatment response	ANN	N/A	N/A	proteomic	Mian et al, 2003
breast	survivability	clustering	statistics	0	clinical	Seker et al, 2003
breast	survivability	fuzzy logic	statistics	N/A	proteomic	Seker et al, 2002
breast	survivability	SVM	N/A	N/A	clinical	Lee et al, 2000
breast	recurrence	ANN	expert	5	mixed	De Laurentiis et al, 1999
breast	survivability	ANN	statistics	1	clinical	Lundin et al, 1999
breast	recurrence	ANN	statistics	23	mixed	Marchevsky et al, 1999
breast	recurrence	ANN	N/A	N/A	clinical	Naguib et al, 1999
breast	survivability	ANN	N/A	N/A	clinical	Street, 1998
breast	survivability	ANN	expert	5	clinical	Burke et al, 1997
breast	recurrence	ANN	statistics	N/A	mixed	Mariani et al, 1997
breast	recurrence	ANN	expert	10	clinical	Naguib et al, 1997
cervical	survivability	ANN	N/A	N/A	mixed	Ochi et al, 2002
colorectal	recurrence	ANN	statistics	12	clinical	Grumett et al, 2003
colorectal	survivability	ANN	statistics	9	clinical	Snow et al, 2001
colorectal	survivability	clustering	N/A	N/A	clinical	Hamilton et al, 1999
colorectal	recurrence	ANN	statistics	9	mixed	Singson et al, 1999
colorectal	survivability	ANN	expert	11	clinical	Bottaci et al, 1997
esophageal	treatment response	SVM	N/A	N/A	proteomic	Hayashida et al, 2005
esophageal	survivability	ANN	statistics	3	clinical	Sato et al, 2005
leukemia	recurrence	decision tree	N/A	N/A	proteomic	Masic et al, 1998
liver	recurrence	ANN	statistics	25	genomic	Rodriguez-Luna et al, 2005
liver	recurrence	SVM	N/A	N/A	genomic	Iizuka et al, 2003
liver	susceptibility	ANN	statistics	-2	clinical	Kim et al, 2003
liver	survivability	ANN	N/A	N/A	clinical	Hamamoto et al, 1995
lung	survivability	ANN	N/A	N/A	clinical	Santos-Garcia et al, 2004
lung	survivability	ANN	statistics	9	mixed	Hanai et al, 2003
lung	survivability	ANN	N/A	N/A	mixed	Hsia et al, 2003
lung	survivability	ANN	statistics	N/A	mixed	Marchevsky et al, 1998
lung	survivability	ANN	N/A	N/A	clinical	Jefferson et al, 1997
lymphoma	survivability	ANN	statistics	22	genomic	Ando et al, 2003
lymphoma	survivability	ANN	expert	10	mixed	Futschik et al, 2003
lymphoma	survivability	ANN	N/A	N/A	genomic	O'Neill and Song, 2003
lymphoma	survivability	ANN	expert	N/A	genomic	Ando et al, 2002
lymphoma	survivability	clustering	N/A	N/A	genomic	Shipp et al, 2002
head/neck	survivability	ANN	statistics	11	clinical	Bryce et al, 1998
neck	treatment response	ANN	N/A	N/A	clinical	Drago et al, 2002
ocular	survivability	SVM	N/A	N/A	genomic	Ehlers and Harbour, 2005
osteosarcoma	treatment response	SVM	N/A	N/A	genomic	Man et al, 2005
pleural mesothelioma	survivability	clustering	N/A	N/A	genomic	Pass et al, 2004
prostate	treatment response	ANN	N/A	N/A	mixed	Michael et al, 2005
prostate	recurrence	ANN	statistics	0	clinical	Porter et al, 2005
prostate	treatment response	ANN	N/A	N/A	clinical	Gulliford et al, 2004
prostate	recurrence	ANN	statistics	16	mixed	Poulakis et al, 2004a
prostate	recurrence	ANN	statistics	11	mixed	Poulakis et al, 2004b
prostate	recurrence	SVM	statistics	6	clinical	Teverovskiy et al, 2004
prostate	recurrence	ANN	statistics	0	clinical	Kattan, 2003
prostate	recurrence	genetic algorithm	N/A	N/A	mixed	Tewari et al, 2001
prostate	recurrence	ANN	statistics	0	clinical	Ziada et al, 2001
prostate	susceptibility	decision tree	N/A	N/A	clinical	Crawford et al, 2000
prostate	recurrence	ANN	statistics	13	clinical	Han et al, 2000
prostate	treatment response	ANN	N/A	N/A	proteomic	Murphy et al, 2000
prostate	recurrence	naïve Bayes	statistics	1	clinical	Zupan et al, 2000
prostate	recurrence	ANN	N/A	N/A	clinical	Mattfeldt et al, 1999
prostate	recurrence	ANN	statistics	17	clinical	Potter et al, 1999
prostate	recurrence	ANN	N/A	N/A	mixed	Naguib et al, 1998
skin	survivability	ANN	expert	14	clinical	Kaiserman et al, 2005
skin	recurrence	ANN	expert	27	proteomic	Mian et al, 2005
skin	survivability	ANN	expert	0	clinical	Taktak et al, 2004
skin	survivability	genetic algorithm	N/A	N/A	clinical	Sierra and Larranga, 1998
stomach	recurrence	ANN	expert	28	clinical	Bollschweiler et al, 2004
throat	recurrence	fuzzy logic	N/A	N/A	clinical	Nagata et al, 2005
throat	survivability	decision tree	statistics	0	genomic	Kan et al, 2004
thoracic	treatment response	ANN	N/A	N/A	proteomic	Seiwerth et al, 2000
thyroid	survivability	decision tree	statistics	N/A	clinical	Su et al, 2005
thyroid	survivability	decision tree	statistics	N/A	clinical	Kukar et al, 1997
tropho-	survivability	genetic algorithm	N/A	N/A	clinical	Marvin et al, blastic 1999

2.2.1 A Model Which Predicts Cancer Susceptibility

In this study Ayer, T., Alagoz, O. et al. (2010) build a model which classifies tumors as either malignant or benign among breast cancer patients. Artificial Neural Networks were used to complete the task with a large degree of success. In fact the model can distinguish malignant or benign tumors with a 96.5% accuracy. It was built with a large number of hidden layers to better generalize data. Thousands of mammographic records were supplied to make the model as accurate as possible. This model proved to be more consistent, effective and less prone to error than most pathologists.

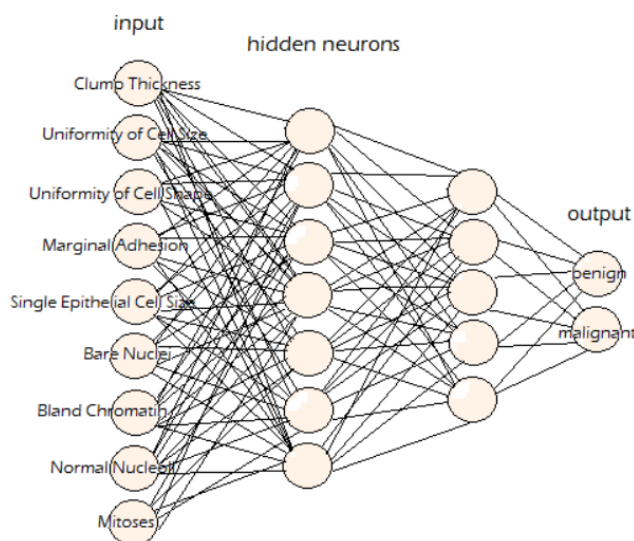


Figure 2.2: Example of Artificial Neural Networks.⁶

2.2.2 A Model Which Predicts Cancer Prognosis

Artificial neural networks (ANNs) and decision trees (DTs) have been used in cancer detection and diagnosis for over 20 years.

Only in the recent years have researchers began to apply machine learning towards cancer prediction/prognosis and risk factors associated with developing cancer. For this reason there is a very limited amount of literature in that field, approximately 120 research papers.

Having said that the use of machine learning in cancer prediction/prognosis is rapidly growing, with the number of papers increasing by 25% every year.

⁵Joseph A. Cruz, D., 2020. Applications Of Machine Learning In Cancer Prediction And Prognosis. PubMed Central (PMC). Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2675494/>.

⁶Sayed, S., 2018. Machine Learning Is The Future Of Cancer Prediction. Available at: <https://towardsdatascience.com/machine-learning-is-the-future-of-cancer-prediction-e4d28e7e6dfa>.

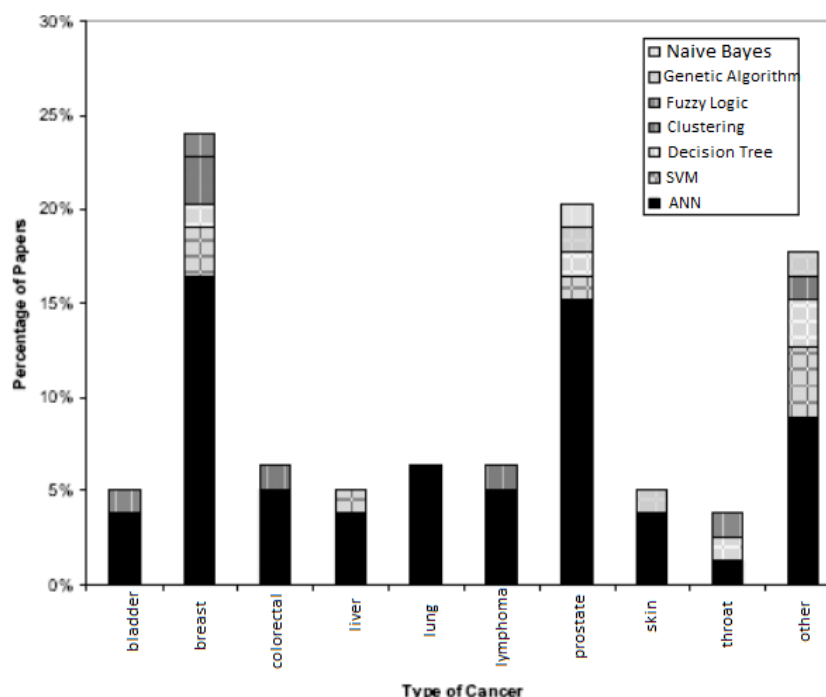


Figure 2.3: Type of Cancer vs Amount of Research Papers and Algorithms Used.⁵

As seen in the figure above, there is strong bias among scientists to use machine learning towards predicting outcomes or risks associated with breast (24%) and prostate (20%) cancer. This is due to the fact that these cancers have a high frequency of occurrence in Europe and North America. However, it appears machine learning has been successful in predicting outcomes or risks in many different types of cancer. A conclusion can be made that some machine learning methods can be applied to cancer prediction/prognosis.

The figure above also illustrates that almost 70% research studies use neural networks as the main predictor. Support vector machines come second with only 9%, while clustering and decision trees each account for about 6%.

Cruz, J. A. and Wishart, D. S. (2007) reviewed multiple applications of machine learning in cancer prediction and prognosis. One of the papers reviewed used single nucleotide polymorphism (SNP) profiles of steroid metabolizing enzymes (CYP450s) to develop a model which would predict the occurrence of “spontaneous” breast cancer.

The hypothesis was that certain combinations of steroid-metabolism gene SNPs would lead to the increased accumulation of toxins or hormones in breast tissue resulting in a higher risk for breast cancer.⁷ The researchers collected SNP data from 63 patients with breast cancer and 74 patients without breast cancer.

⁷Cruz JA, Wishart DS. Applications of machine learning in cancer prediction and prognosis. *Cancer Inform.* 2007;2:59–77. Published 2007 Feb 11.

They then used several methods to reduce the sample-per-feature ratio which allowed the study to avoid falling victim to the “curse of dimensionality”, which you can read more about in section 2.6.3.

The researchers then investigated multiple machine learning algorithms to find the optimal classifier, amongst these: naive bayes, several decision tree models and a sophisticated support vector machine (SVM).

Finally, extensive level of cross validation was performed. Each model was validated in minimum 3 ways. The training of the models was assessed and monitored with 20-fold cross-validation. A bootstrap resampling method was employed by performing the cross-validation 5 times and averaging the results thus minimising bias.

The SVM achieved the highest accuracy of 69% while the decision tree classifier and naive bayes classifier achieved an accuracy of 68% and 67% respectively. These results are approximately 23–25% better than chance. Results presented in this study have been replicated with a similar study of another 200 individuals.

This study perfectly illustrates how careful data preparation, thoughtful implementation, suitable data selection and detailed validation of multiple machine learners can produce an exceptional and accurate cancer-risk prediction model.

2.2.3 A Model Which Predicts Cancer Survival rates

Machine learning has also been used to predict cancer survival rates. Kourou K, Exarchos P. et al. created a model in 2014, using a dataset of 162,500 records with 16 features.

Based on the features such as the size of the tumor and the age of the patient, the model created was able to classify if the patient survived or not. The researchers used SVM’s, ANN’s and semi-supervised learning (SSL: a mix between supervised and unsupervised learning).

SSL’s was the most successful approach with an accuracy rate of 71%. Another, very similar study also used ANN’s to predict the survival rate of patients suffering from lung cancer. This model had an accuracy rate of 83%, thus showing that ANN’s should definitely be an algorithm to consider when building similar models.

Another study worth mentioning in this section, conducted by Montazeri Mitra, Montazeri Mohadeseh et al. (2016) dealt with the prediction of the survival rate for various types of breast cancer.

The researchers used a dataset with eight attributes and 900 records of which 876 (97.3%) patients were female and only 24 (2.7%) patients were male. They then compared naive bayes (NB), trees random forest (TRF), 1-Nearest Neighbor (1NN), Support Vector Machine (SVM), RBF Network (RBFN), and Multilayer Perceptron (MLP) machine learning algorithms.

Evaluation was performed using 10-cross fold technique and the performance measured with respect to accuracy, precision, sensitivity, specificity, and area under ROC curve. Out of 900 patients, 803 patients were alive and 97 patients were dead.

In this study, Trees Random Forest achieved the highest accuracy of 96%. The researchers concluded that this model is recommended as a useful tool for breast cancer survival prediction as well as medical decision making.

2.2.4 A Model Which Predicts Cancer Recurrence

Machine learning also helped to predict the recurrence of oral cancer after the total remission of cancer patients, which has been documented in another study conducted by Xie X, Hu Y and Jing Ch (2017). Clinical, imaging and genomic data was collected from 86 patients for this model. Careful feature selection decreased the model's features from over 110 to just 30. This greatly reduced noise and overfitting, thus reducing bias and making the model much more accurate.

The researchers used BN's, ANN's, SVM's, DT's and RF's to classify patient data into those with cancer relapses and those without. The model correctly predicted all patients using feature selected and BN's. Although the accuracy was extremely high, the model had a really small dataset of only 86 patients.

2.3 Study Review: Diagnosis of Hepatocellular Carcinoma

One of the studies which deserves a closer look is a study conducted by Masaya Sato published on the 30th of May, 2019.
(Sato M., Morimoto K., Kajihara S., 2019)

Objective

Here the researchers aimed to create a model for the prediction of hepatocellular carcinoma (HCC). They used real-world data obtained during clinical practice. They developed a framework which established the best classifier and using the grid-search method found the optimal hyperparameters for that classifier.

Data Collection

From all the patients who visited Tokyo Hospital between January 1997 and May 2016, 4242 patients were chosen (1311 HCC patients and 2931 non-HCC patients). From these only patients for whom information on AFP, AFP-L3, DCP, AST, ALT, platelet count, alkaline phosphatase (ALP), gamma-glutamyl transferase (GGT), albumin, TB, age, sex, height, body weight, hepatitis B surface (HBs) antigen, and hepatitis C virus (HCV) antibody status were available were selected.

This left the researchers with a dataset of 1582 records, containing 539 patients with HCC and 1043 patients without HCC.

The initial diagnosis was performed using dynamic computed tomography (CT) imaging, with hyper-attenuation during the arterial phase and washout during the late phase regarded as a definite sign of HCC. If a certain diagnosis of HCC was not made using CT, an ultrasound-guided tumor biopsy was conducted.

Feature Selection

As said before a HCC diagnosis can not be based entirely on the AFP test. This is why multiple biomarkers have to be combined in order to improve the accuracy of the model. These include des-gamma-carboxyprothrombin (DCP), Lens culinaris agglutinin-reactive fraction of AFP (AFP-L3), biomarkers of liver inflammation such as aspartate aminotransferase (AST) and alanine aminotransferase (ALT), fibrosis (platelet count), and biomarkers of liver function, total bilirubin (TB) and albumin along with the hepatitis virus status.

Algorithm Selection

The researchers used a linear logistic regression model for the linear classification and support vector machines using an RBF kernel, gradient boosting, random forests, neural networks, and deep learning for non-linear classification model.

Evaluation

To evaluate the accuracy of the model, the researchers split the data into 3 groups at random:

- a) Training set (80%), used to teach the model.
- b) Development set, used for parameter tuning.
- c) Test set, used to evaluate the efficiency of every algorithm and determine the predictive accuracy of the model. As in previous studies the area under the curve (AUC) was considered as the ability to diagnose HCC.

Results

The number of patients who were male, had a high HCV antibody-positivity, and HBs antigen-negativity was significantly higher among the HCC patients. The levels of AFP, AFP-L3, DCP, AST, ALP, GGT, and TB, and the patient age were also much higher among the HCC patients. On the other hand the level of ALT, platelet count, and albumin level were lower.

Using the optimal hyperparameters, gradient boosting gave the highest predictive accuracy for the presence of HCC of 87.34% and produced an area under the curve (AUC) of 0.94.

The most important features for HCC prediction were: age, three tumor markers and albumin level. An ROC analysis showed that the AUC, sensitivity, and specificity for this optimal classifier were 94%, 93.27%, and 75.93%, respectively.

Conclusion

Model fitting is important for a successful predictive method.

If the data is linearly separable, a linear model will fit the data very accurately. On the other hand if the data is not linearly separable, a non-linear model will fit the data much better.

Therefore, the algorithms have to be selected in regard to the data itself. Identifying the optimal learning parameters for every algorithm is very important and should be done properly using a grid search method.

2.4 Study Review: Hepatocellular Carcinoma Survival Prediction

The Hepatocellular Carcinoma Survival Prediction Study performed by a Satish Chandra Reddy Nandipati, Haziqah Shamsudin and Chew XinYing in the School of Computer Sciences, University Sains Malaysia, Pulau Pinang, Malaysia is particularly important as not only is it exactly what this research is about but it also uses the exact same dataset.

This study provided shows exactly what approach has been taken before, presents results, but also provides a set bench mark for this research.

(Nandipati SCR, Shamsudin H., et al., 2019)

This study was used as a stepping stone to achieve better and more accurate results. It is also a study which was published in the Amity Journal of Computational Sciences (AJCS), meaning that this current research could potentially also be published in a top venue if the results proved to be more accurate.

Objective

The objectives of this study was to create a model using Rapid miner version 9.2, which would accurately predict the survival rate of patients with the HCC disease.

Data Sources

The authors used the exact same dataset as is used in this research.

They used the complete balanced dataset in which the missing values were imputed using KNN (K=1) and HEOM distance. The balance nature of the dataset was achieved using SMOTE (k = 3) with oversampling method.

Methodology

Normalization was used in the data preprocessing stage.

Feature selection was performed using forward election (method = Naïve Bayes) and backward elimination (method = Naïve Bayes and decision tree).

Next the data was split into training/testing sets using a 70%/30% split.

Then the model was trained using seven different algorithms:

K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB) and Multilayer Perception (MLP), Bagging (method = decision tree) and Adaboost (method = decision tree). Default values were used for all the parameters for all algorithms.

The models were evaluated using 10-fold cross validation.

Results

The researchers initially used all 49 features and seven different machine learning algorithms. SVM achieved the highest accuracy of 81.81% with random forest coming close second achieving accuracy of 79.67%.

These results can be seen in table 2.2.

The average accuracy from all seven models was 75.19%.

Later, the authors used 7 of the features selected by forward selection and once again seven different algorithms. Naive Bayes achieved the highest accuracy of 74.90% followed by SVM with 74.10%.

The average accuracy this time around was 71.93%.

These results are presented in table 2.3.

Table 2.3: Results when 49 features were used.⁸

Algorithms	Accuracy	Precision	Recall
KNN	74.24	79.37	67.57
SVM	81.81	79.27	87.84
RF	79.67	78.48	83.78
NB	72.41	75	68.92
Auto MLP	76.81	77.33	78.38
Bagging	63.86	63.41	70.27
AdaBoost	77.57	77.63	79.73
Average	75.19	75.78	76.64

Table 2.4: Results when 7 features were used.⁸

Algorithms	Accuracy	Precision	Recall
KNN	72.76	71.60	78.38
SVM	74.10	71.76	82.43
RF	72.71	72.15	77.03
NB	74.90	73.75	79.73
Auto MLP	72.10	70.73	78.38
Bagging	69.86	67.82	79.73
AdaBoost	67.09	66.67	75.68
Average	71.93	70.64	75.765

⁸Amity.edu. n.d. Available at: https://amity.edu/UserFiles/aijem/512019_V03_I01_P012-016.pdf.

2.5 Description of Algorithms

By learning from the studies outlined above a few conclusions were drawn. Clearly, some algorithms outperformed others. However, an assumption that some algorithm would always outperform other can not be made. This is because the way in which data-processing is done always impacts the choice of algorithm. Some algorithms such as Multi-layer Perception may perform better when there are many features and records present, whereas simple Logistic Regression might perform better when a dataset is composed of not so many features. Thus, few algorithms were chosen for further research and several were eliminated based on their performance in previous studies.

2.5.1 Logistic Regression

Logistic Regression is a classification algorithm. Like all machine learning algorithms it is based on the concept of probability. Logistic Regression is also considered a Linear Regression algorithm. However the cost function of the Linear Regression algorithm, as some call it the ‘Sigmoid function’ or also known as the ‘logistic function’, is more complex than the cost function of simple Linear Regression.

To represent predicted values as probabilities, the Sigmoid function has to be used. This function maps real values into values between 0 and 1. Based on this probability the algorithm then gives us a set of outputs, classifying each record as a 0 or as a 1.

In this model, the cost function represents a optimization objective. In other words, after the cost function has been created, it is minimized and therefore a more accurate model with minimum error emerges. The cost value is reduced by using Gradient Descent. Gradient Descent is a technique which involves iteratively moving in the direction of steepest descent as defined by the negative of the gradient and continuously updating the parameters.

2.5.2 Support Vector Machine

Support Vector Machine is another ML classifier. It is currently a very popular solution to many classification problems.

It considers the boundary between two outputs and uses these as ‘support vectors’. This boundary is also called the hyperplane.

The distance between the hyperplane and the nearest data point from either set is known as the margin. The output for new data is calculated by minimising the overall margin distance.

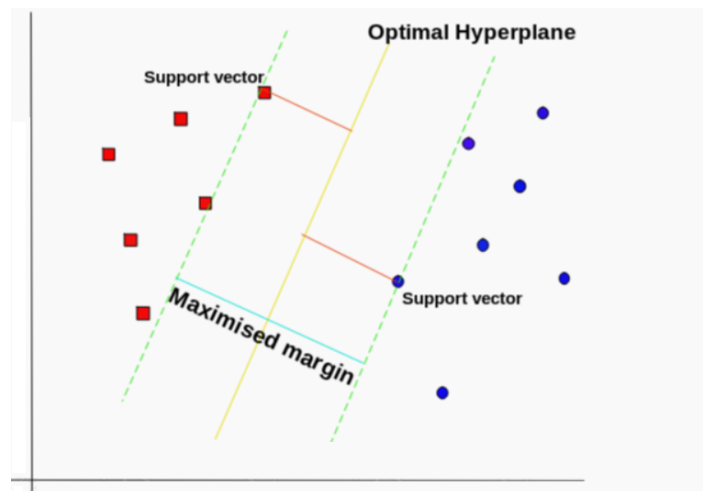


Figure 2.4: Support vectors, hyperplane and margins.⁹

If there is no clear hyperplane due to the complexity of the data, the SVM uses kernelling. Kernelling means mapping data points to a higher dimension thus creating a more separable data and creating a hyperplane. This allows SVM to generalise well, which can be both good and bad.

New data inputs which may have similar outputs but quite different inputs may have a much greater chance of being classified as the same.

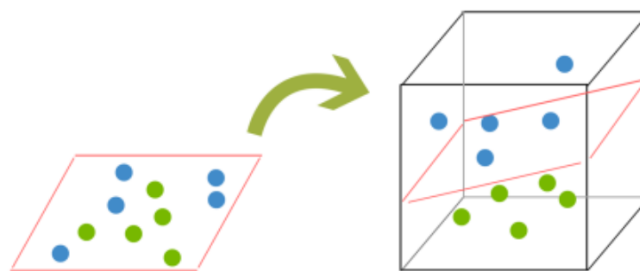


Figure 2.5: Kernelling or “The Kernel Trick”.¹⁰

⁹Support Vector Machines(SVM) — An Overview. Available at: <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>.

2.5.3 Random Forest

Random Forest is an ensemble algorithm. Ensemble algorithms use multiple learning algorithms and averages their results to obtain better predictive accuracy than could be obtained from any of the used learning algorithms alone.

It is also a “bagging” algorithm. Bagging ensemble models are algorithms where all individual algorithms used are completely independent of each other, only their outputs are combined to acquire a more accurate performance.

Hence Random Forest is just a combination of individual models which are Decision Trees. Firstly a Decision Tree algorithm creates a tree with the “leaves” being the output. For every record in the dataset, the algorithm moves through the tree making a decision at every node until it arrives at one of the leaves, hence classifying each input .

The Decision Tree Algorithm chooses the features which are the most correlated to the output and prioritises them when creating decision nodes. If one or a few features are very strong predictors for the target output, these features will be selected in many of the trees. The output of each Decision Tree is then combined and the overall output calculated.

Initial research has shown that Random Forests are less susceptible to overfitting and are generally more accurate than a single Decision Tree, the extent of which increases with the size of the forest (Ho, 1995).

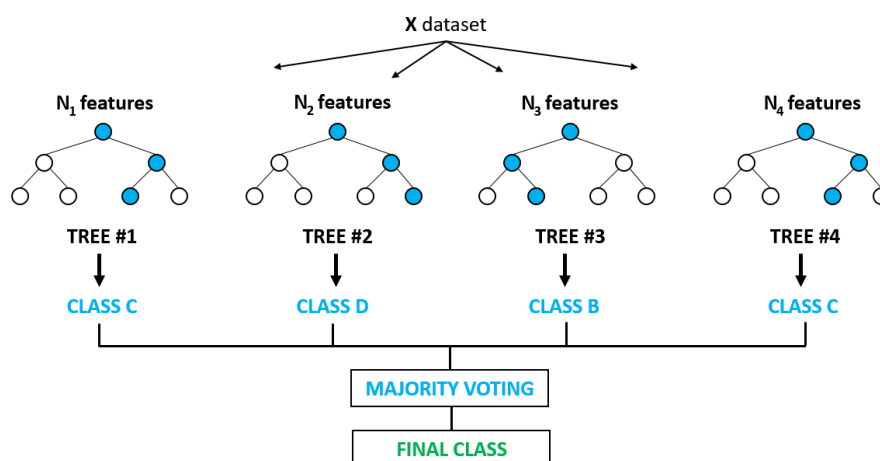


Figure 2.6: Representation of Random Forest.¹¹

¹⁰KDnuggets. n.d. Support Vector Machines: A Simple Explanation - Kdnuggets. Available at: <https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html>.

¹¹Medium. n.d. Applying Random Forest (Classification) — Machine Learning Algorithm From Scratch With Real. Available at: <https://medium.com/@ar.ingenious/applying-random-forest-classification-machine-learning-algorithm-from-scratch-with-real-24ff198a1c57>.

2.5.4 Multi-layer Perceptron

The Multilayer Perceptron is just a segment of neural network algorithms. Neural Networks attempt to mimic the way in which the human brain functions. The architecture of these algorithms replicates the structure of the human brain. Features are seen as neurons, each connected with weighted links. In simple, linearly separable problems a single neuron can provide a solution, however for complex problems multiple layers of neurons are needed.

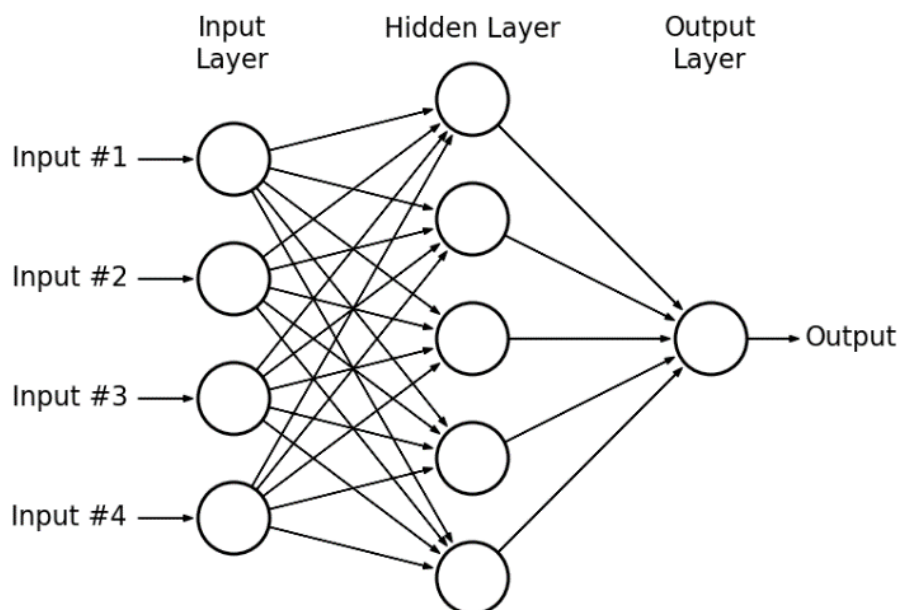


Figure 2.7: Multilayer-Perceptron-Network.¹²

The first layer also called the input layer contains the first set of neurons. Each neuron in this layer represents one of each numerical input features or one of each possible value of a categorical feature.

The layers after that are called hidden layers and each contains a number of neurons. The number of hidden layers and the number of neurons within these layers changes with every model created. The optimal number of these can only be determined through trial and error.

The last layer is called the output layer and contains a single neuron when dealing with a regression problem.

¹² Available at: https://www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065.

The network learns through the process called “backpropagation”. The input data moves through the network all the way from input layer through the hidden layers to the output layer. The predicted output is then compared with the correct output. The error is then propagated back from the output layer to the input layer, updating the weights on the links which connect the neurons along the way.

This process is then repeated for a number of iterations, called epochs. The weights are adjusted after every epoch. After every adjustment the result should be closer to the correct output.

2.5.5 K-Nearest Neighbors

The K-Nearest Neighbour algorithm has already been used since the 1970’s. The classification of an instance is done by a majority vote, based on its neighbors. The instance is assigned the class which its K-number of neighbours have. For example, if $K = 1$, then the case is simply assigned to the class of its nearest neighbor. The nearest neighbours are found using a distance function. However, depending on the data different distance functions have to be used. When choosing the optimal value for K it is advised to inspect the data thoroughly first. One could assume that a large K value would be precise as it would reduce the overall noise of the data but there is no guarantee. For example, if the data is linearly separable and the instance to be predicted is close to the boundary, neighbours on the opposite side could affect the classification. Historically, K between 3-10 seems to be optimal.

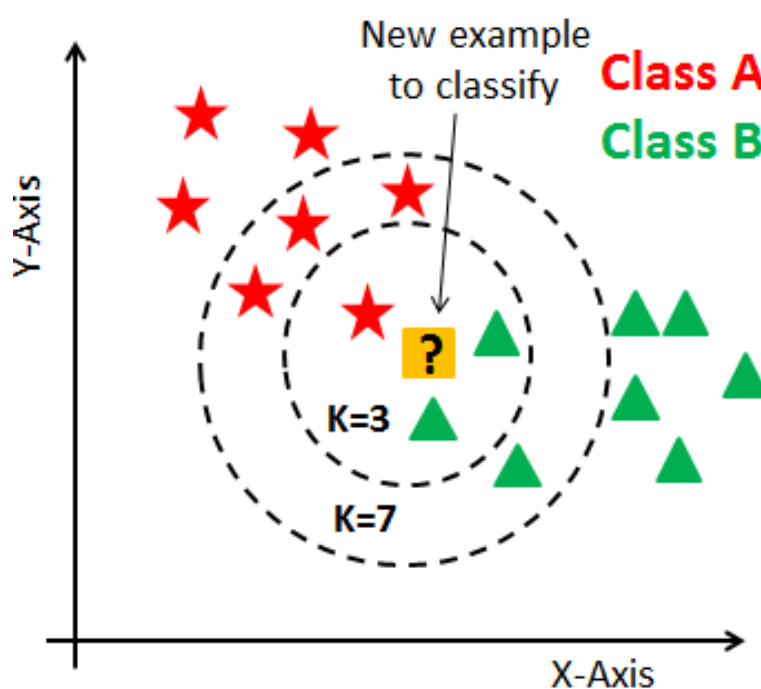


Figure 2.8: K-Nearest Neighbors.¹³

2.6 Common Machine Learning Challenges

Several challenges have been met over the course of the research. The challenges involved data collection and overall dilemmas which come with the development of machine learning models. All the challenges are outlined and explained below.

2.6.1 Lack Of Data

As mentioned before, there are only a few open data sources. These sources contain very limited amount of medical data due to the fact that medical data is extremely confidential. Patients rather not share their medical records even if their anonymity is guaranteed.

One of the reasons for this is the fact that even the most protected data can be hacked and made public or even sold. In 2019 researchers from WizCase discovered nine separate, unsecured medical websites leaking sensitive data from millions of patients around the world, including health information, Social Security numbers, and other sensitive data.

Unfortunately, most businesses do not see value in devising predictive models for cancer, as they can profit much more from selling drugs for conditions related to cancer. This means that the only reason for companies to develop these models, would be to boost their public image or receive government funding, hence indirectly profit financially.

This means that continuous treatment of cancer is something many companies profit from. Some cancer patients face out-of-pocket costs of nearly \$12,000 a year for one drug. It has been reported that 2014, cancer patients paid \$4 billion out-of-pocket for cancer treatment, with newly approved cancer drugs costing an average of \$10,000 per month, with some even as high as \$30,000 per month. It is clear why companies would be discouraged to permanently fix the problem of cancer or even to develop applications, new diagnosis techniques or treatment methods.

¹³Navlani, A., 2018. KNN Classification Using Scikit-Learn. DataCamp Community. Available at: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>.

2.6.2 Bias vs. Variance

The prediction error of any machine learning algorithm can be defined by three separate parts:

a) Irreducible Error - this error cannot be reduced regardless of what algorithm is used. It is the error which emerges when the problem is being framed. It may be caused by factors like unknown variables that influence the mapping of the input variables to the output variable.

b) Bias Error - Bias is the difference between the models predicted output and the actual observed output, and it can sometimes appear to trend in one direction. Linear algorithms tend to have a high bias making them fast to learn and easier to understand but generally less flexible. Bias is a result of attempting to learn too many features.

c) Variance Error - Variance is the amount that the estimate of the target function will change if different training data was used. The target function is estimated from the training data by a machine learning algorithm, so we should expect the algorithm to have some variance. Ideally, it should not change too much from one training dataset to the next, meaning that the algorithm is good at picking out the hidden underlying mapping between the inputs and the output variables. Variance usually occurs as a result of putting too much emphasis on outliers that may be unique to a dataset.

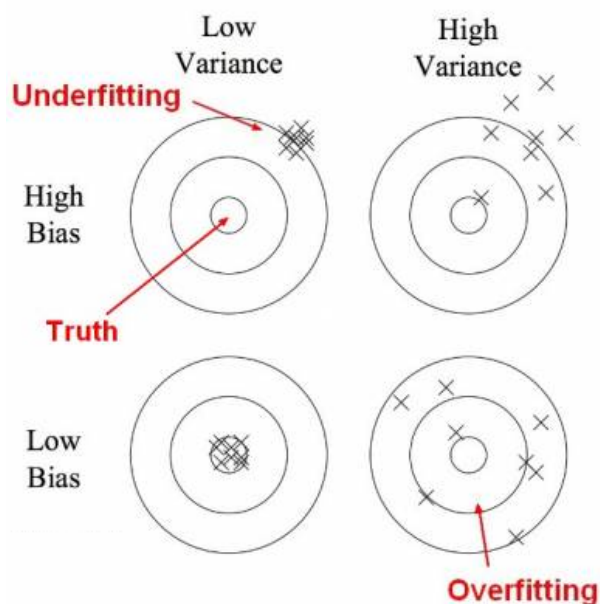


Figure 2.9: Bias vs. Variance.¹⁴

¹⁴Understanding The Bias-Variance Tradeoff. Available at: <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>.

2.6.3 The Curse of Dimensionality

Dimensionality refers to the number of features contained within the dataset. One could assume that a greater number of features means the algorithm will perform better predictions due to the fact it has more information to work with. However some features may have a very low correlation with the target variable. In this case the algorithm will try and use the feature and somehow fit it into the model. This can lead to overfitting and cause very inaccurate results. In addition to this more features will also add to the computational complexity of the model.

2.6.4 Overfitting vs. Underfitting

Overfitting - happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model's ability to generalize.

Underfitting - happens when a machine learning model is not complex enough to accurately capture relationships between a dataset's features and a target variable. An underfitted model results in problematic or erroneous outcomes on new data, or data that it wasn't trained on, and often performs poorly even on training data.

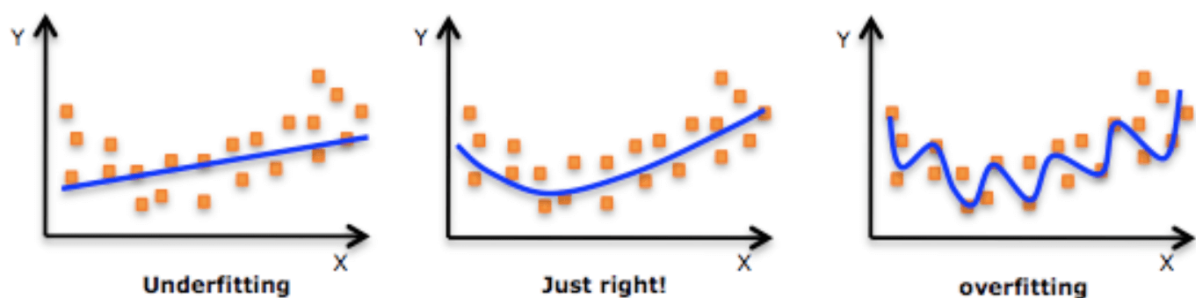


Figure 2.10: Overfitting vs. Underfitting.¹⁵

¹⁵Model Fit: Underfitting Vs. Overfitting - Amazon Machine Learning. Available at: <https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html>.

2.6.5 Computational Complexity

All software applications have to be designed in such a way that the task they perform is worth sacrificing the resources the application is using. To use a simple example, a computer clock application should not take up 100% of the CPU, as the task its performing is extremely simple.

Machine learning models often benefit from acquiring more resources such as time or computational power. However there comes a point where the amount of resources provided does not match the benefits attained. For example, gaining 1% higher model accuracy may not be worth allowing the model to learn for 10 days at full computational power. The balance is usually found due to past experience and trial and error.

2.7 Summary

This chapter concludes with a short summary of what has been learned.

Firstly, a quick description of artificial intelligence and machine learning gave more insight into how the two terms are used, what do they mean, how are they connected and how they impact our world today. Next, a description and review of the the state of the art gave a deeper understanding of the possibilities machine learning and artificial intelligence both offer.

After that, a thorough review of past research on the topic of cancer and machine learning classification provided a clearer path which should be taken in the proceeding stages of this research. The previous research allowed for a more clear choice of machine learning algorithms based on their performance in previous studies, these algorithms were then explained in greater detail.

Finally, a short description of the challenges which may arise, gave more insight into what a researcher should focus on when conducting a study such as this. This literature review, and knowledge acquired, allowed for a much quicker, safer and more successful implementation process.

Chapter 3

Design and Implementation

The literature review provided clear design choices which should be taken during the implementation process. These decisions, and the reasoning behind them, are presented and explained in this chapter.

The chapter begins with technology selection section, where the contrast between different technologies is outlined. Then the chapter gives a quick evaluation of the data-collection process. Finally, it finishes with an extensive description of the data-preprocessing and model training process.

All of the stages mentioned above and all the processes within these stages intend to maximise the F1 and F2 scores of predictions made for:

- (a) Liver disease diagnosis
- (b) Hepatocellular Carcinoma Survival

3.1 Technology Selection

The first design decision to be made was the choice of a programming language suitable for a machine learning application. Python seemed to be the most obvious choice for almost any data science or machine learning application thanks to its flexibility, simplicity, the number of different libraries which Python is home to, and finally the way in which it deals with errors by providing detailed and very helpful error messages.

Python was also the machine learning language which the author had the most experience working with. Python therefore allows users who are completely new to data sciences to quickly learn and develop their own machine learning models. Its simplicity also means that less time is spent fixing errors and more time is spent increasing the performance of the models.

R is also a well know programming language often used for the development of machine learning models. Comparing the two languages and their capabilities resulted in a tie. R has over 5000 various libraries while Python is home to some very useful packages like Pandas, NumPy, SciPy, Scikit Learn and Matplotlib. However, R has an amazing visualizations package called ggplot2, which could've been extremely useful and possibly cut out the need for importing two separate graphing libraries. In the end Python was selected as R would've requires more initial studying effort which was not possible under the projects time constraint.

The second design decision concerned the selection of libraries which would be useful and optimal for this application. As said before Python possess a very broad amount of libraries, and in that a vast number of libraries which deal with machine learning implementation.

Scikit-learn was the library chosen as the main machine learning library partially due to the fact that the author had previous experience working with this library. Scikit-learn possesses a wide variety of machine learning algorithms for both classification and regression.

Tenserflow was another contender, however it deals mostly with neural networks, which weren't really the focus of this study.

All the features provided by the Scikit-learn library allow for quick implementation and evaluation of the models created, but also for quick and easy comparison between the models themselves, which was critical for this study.¹⁶

The Pandas library was chosen for data manipulation. This library makes it extremely easy to manipulate data in all kinds of way.

Not only that, it also allows for quick and clean presentation of data as all of the data is stored in a dataframe format.¹⁷

Matplotlib was the library chosen to create graphical representation of the data and the results.¹⁸

The Seaborn library was also chosen for this purpose. The Seaborn graphs are very clean and modern, whereas Matplotlib excels in different areas. These libraries proved to be ideal for the success of the study.¹⁹

The file format known as Comma-Separated Values (CSV) is a delimited text which uses commas to separate values and lines to separate record/instances. Each line of the file is a data record.

Each record consists of one or more fields, separated by commas.

This format was chosen for data storage.

Pandas can easily read in the CSV format, and then convert the file to a singular to even multiple different dataframes which can then be manipulated or merged together. The original data was also saved in this file type which made it very easy for the author to use.

Due to the fact that the data was collected from an open source, different, more secure file type was not needed.

¹⁶<https://scikit-learn.org/stable/>

¹⁷<https://pandas.pydata.org/>

¹⁸<https://matplotlib.org/>

¹⁹<https://seaborn.pydata.org/>

3.2 Data Collection

3.2.1 HCC Survival Dataset

The Hepatocellular Carcinoma Survival Dataset was easily acquired from UCI - Machine Learning Repository, a dataset storage platform which contains hundreds of open source datasets. The HCC dataset was originally collected at a University Hospital in Portugal and donated to UCI by Miriam Seoane Santos and Pedro Henriques Abreu from the University of Coimbra and Armando Carvalho and Adélia Simão from the Hospital and University Centre of Coimbra.

The dataset contains real clinical data of 165 patients diagnosed with HCC. Obtaining the data did not require signing up to the platform, only downloading a zip file which contained the CSV file with the dataset and other files which described the dataset and provided additional information.

This dataset can be acquired from:

<https://archive.ics.uci.edu/ml/datasets/HCC+Survival>

3.2.2 Liver Disease Dataset

The Liver Disease Dataset Dataset was acquired from Kaggle, the data science and machine learning community platform. This dataset was originally collected from North East of Andhra Pradesh, India. This dataset contains 416 liver patient records and 167 non liver patient records and also possesses a certain gender imbalance as it contains 441 male patient records and 142 female patient records. Obtaining this data simply meant signing up for the Kaggle service and downloading a zip file which contained the CSV file. This dataset can be acquired from:

<https://www.kaggle.com/uciml/indian-liver-patient-records>

3.2.3 Handling The Unbalanced Datasets

Both the HCC dataset and the Liver Disease dataset collected were not perfect. For example, the HCC dataset possessed a certain degree of class imbalance as there were 63 patients labeled as “dies” and 102 as “lives”. Using a machine learning algorithm on a dataset with such an imbalance where one class clearly dominates the other can be quite problematic.

In some cases the dominance could be so great that the algorithm could classify every value in the test set as the same class and still be seen as quite accurate.

There are 4 ways of dealing with the imbalance problem:

- a) Synthesis of new minority class instances
- b) Over-sampling of minority class
- c) Under-sampling of majority class
- d) Tweak the cost function in a way that the misclassification of minority instances more is more important than misclassification of majority instances.

The imbalance in these datasets was dealt with through “a) Synthesis of new minority class instances” and specifically using the Synthetic Minority Over-sampling Technique (SMOTE).

SMOTE synthesises new minority instances between existing minority instances. SMOTE basically draws connections between the instances of the minority class and then synthetically generates additional minority instances along these lines.

In more technical terms SMOTE iterates through the real minority instance. After each iteration, one of the K closest minority class neighbours is chosen and a new minority instance is synthesised somewhere between the minority instance and that neighbour.

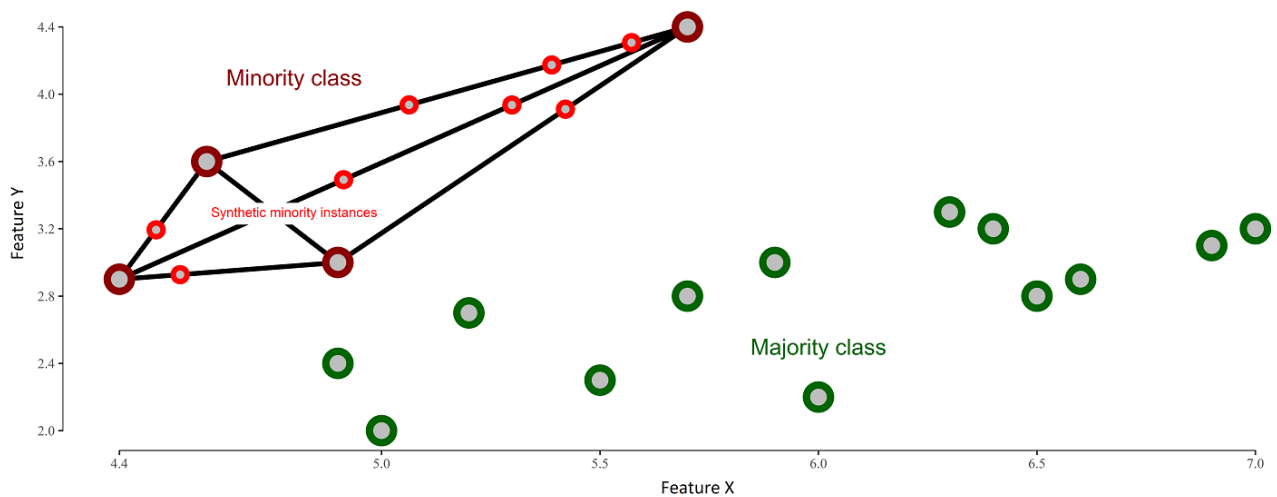


Figure 3.1: Addressing class imbalance via the SMOTE method.²⁰

SMOTE was chosen as although both oversampling and undersampling achieve class balance, they can potentially hinder the learning task: oversampling does not incorporate any new information and may lead to overfitting, while undersampling may remove important examples to the learning step, causing the classifier to miss important concepts.

²⁰Rikunert.com. n.d. SMOTE Explained For Noobs - Synthetic Minority Over-Sampling Technique Line By Line · Rich Data.Available at: <http://rikunert.com/SMOTE.explained>.

3.3 Data Pre-Processing

As mentioned before, this chapter outlines all the pre-processing techniques performed on the two datasets used for model training and the reasoning behind them. This step is sometimes considered the most important in model development and can take up to 80% of the entire ML process (Dasu and Johnson, 2003).

3.3.1 Data Cleaning

Data cleaning involves selecting and performing different procedures on the datasets to detect and correct any records which are incomplete, incorrect, inaccurate or irrelevant within these datasets. Different methods have pros and cons and all of these have to be considered when performing data cleaning. At the end of this step, all incorrect data should have either been removed, corrected, or transformed.

3.3.1.1 Missing Data

Only the HCC Survival dataset contained missing values within features. Table 3.1 presents the percentage of missing values. "Oxygen Saturation %" and "Ferritin" have the highest number of missing values as the values for them are missing in 48% of the records.

Multiple approaches were considered (method (a) below) and tested (methods (b),(c), and (d)) when dealing with missing values. These were:

(a) Dropping Records - This method may be a viable option when the datasets contain vast amounts of records. For example, if a dataset contained 10,000 records and 100 of these contained missing values, dropping these would mean deleting just 1% of your data which usually should not have a huge impact on the model's accuracy of F scores. This was not a viable method for this research as both datasets are quite small. When using this method, the data should be examined thoroughly to ensure vital data the model could learn from, is not deleted.

(b) Imputation Using Mean or Median Values - This method includes calculating the mean or median of the non-missing values in a given column and then replacing the missing values within each column separately and independently from the others. However this method can only be used on numeric data. The cons of this method include: not factoring in the correlations between features and not being very accurate.

(c) Imputation Using k-NN Algorithm - This technique makes use of the k nearest neighbours algorithm. Missing values are replaced based on how closely the record resembles records closest to it. The optimal value for k has to be found through trial and error. However this method is computationally expensive as k-NN works by storing the whole training dataset in memory. Although this method is much more accurate than methods (a) and (b) it also doesn't work for categorical values.

(d) Imputation Using the Most Frequent Values - This was the most successful method out of all the methods mentioned previously. This technique simply finds the most frequent value in each column and replaces all the missing values within that column with that value. This method works with categorical features (strings or numerical representations).

Table 3.1: Percentage of missing values in the HCC Survival dataset.

	Total	Percent
oxygen_saturation_%	79	0.481707
ferritin	79	0.481707
iron	78	0.475610
packs_of_cigarets_per_year	53	0.323171
esophageal_varices	52	0.317073
grams_of_alcohol_per_day	48	0.292683
direct_bilirubin_mg/dL	44	0.268293
smoking	41	0.250000
hepatitis_b_e_antigen	39	0.237805
endemic_countries	39	0.237805
hepatitis_b_core_antibody	24	0.146341
hemochromatosis	23	0.140244
nonalcoholic_steatohepatitis	22	0.134146
major_dimension_of_nodule_cm	20	0.121951
symptoms	18	0.109756
hepatitis_b_surface_antigen	17	0.103659
splenomegaly	15	0.091463
human_immunodeficiency_virus	14	0.085366
portal_hypertension	11	0.067073
total_proteins	11	0.067073
obesity	9	0.054878
hepatitis_c_virus_antibody	9	0.054878
alpha-fetoprotein	8	0.048780
creatinine	7	0.042683
albumin	6	0.036585

3.3.2 Feature Engineering

Feature engineering was the next step in preparing the datasets for the model teaching process. This step extracts the most relevant information from the datasets and formats this information into a structure, from which machine learning algorithms can learn most accurately and efficiently. All of the feature engineering methods performed on the two datasets are outlined bellow.

3.3.2.1 Feature Encoding

Feature encoding means taking a dataset and transforming the values within the features or the dataset itself into a form which a machine learning algorithm would understand.

For example, most machine learning algorithms cannot handle categorical data. Decision Trees is an example of an algorithm which can handle categorical features and therefore many ensemble algorithms make use of Decision Trees. The scikit-learn library however only allows categorical features to be used with a singular Decision Tree and not ensemble algorithms.

Therefore these features have to be transformed into something the algorithms can grasp, while still retaining all of their information.

There is a number of different methods for feature encoding, several of which have been used on the datasets used in this project.

These were:

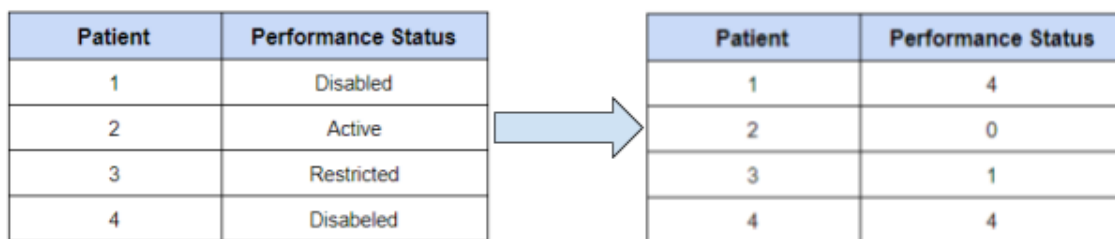
(a) Label Encoding - this method simply takes string values within a feature and assigns them an integer value. In the HCC Survival dataset the columns: ‘performance status’, ‘encephalopathy degree’ and ‘ascites degree’ all contained integer values ranging from 0 to 4, where each integer represented a different condition.

For example, ‘performance status’ was represented as:

[0 = *Active*; 1 = *Restricted*; 2 = *Ambulatory*; 3 = *Selfcare*; 4 = *Disabled*]

If this feature was left unchanged the data would become stratified ie. hierarchy would be added to the data.

For example, the machine learning algorithm could view ‘Disabled’ as something which is ‘higher’ or more positive than ‘Active’, which in reality is not true.



The diagram illustrates the flawed Label Encoding method. It shows two tables connected by a large blue arrow pointing from left to right. The left table has two columns: 'Patient' and 'Performance Status'. It contains four rows of data: Patient 1 is Disabled, Patient 2 is Active, Patient 3 is Restricted, and Patient 4 is Disabled. The right table has the same two columns but with numerical values instead of categorical ones: Patient 1 is 4, Patient 2 is 0, Patient 3 is 1, and Patient 4 is 4. This demonstrates how the categorical hierarchy is lost when converted to integers.

Patient	Performance Status
1	Disabled
2	Active
3	Restricted
4	Disabeled

Patient	Performance Status
1	4
2	0
3	1
4	4

Figure 3.2: Flawed Label Encoding method.

(b) One Hot Encoding - One hot encoding was the most successful method used. This method produces a new boolean feature for each unique value in the column. For example, the ‘performance status’ feature would be transformed into 4 separate features classifying each patient as for example:

[0 = *Active*; 1 = *Restricted*; 0 = *Ambulatory*; 0 = *Selfcare*; 0 = *Disabled*]

meaning a patient is not active, not ambulatory, not in self care, not disabled but is restricted. It is worth noting that when the cardinality of the feature is high, the dimensionality greatly increases. In that case binary encoding should be used. In the case of the features mentioned above the cardinality was so small that the increase in dimensionality did not have an impact on the F scores or accuracy of the results.

Patient	Performance Status			
1	Disabled			
2	Active			
3	Restricted			
4	Disabeled			

Patient	Active	Restricted	Ambulatory	Disabled
1	0	0	0	1
2	1	0	0	0
3	0	1	0	0
4	0	0	0	1

Figure 3.3: One Hot Encoding used on the ‘performance status’ feature.

3.3.2.2 Feature Scaling

The final step in the feature engineering process was feature scaling. The features in the datasets have varying scales. This can cause the machine learning algorithms to put more weight on the features with a bigger scale, which may lead to results which are much more inaccurate than results which could be obtained if all features were considered of equal importance. For example, the scale of the ‘ferritin’ feature in the HCC Survival dataset ranges from 0 nanograms per millilitre to 2230 nanograms per millilitre while the scale of the ‘iron’ feature ranges from 0 micrograms per decilitre to only 224 micrograms per decilitre.

This step is also extremely relevant as almost all of the features in the datasets used were computed using different units of measurement, such as: nanograms per millilitre, grams per litre or in some cases units per litre. These features had to be brought to similar scales via various scaling methods. These were:

(a) Normalization - this method is the most popular amongst all scaling methods. However, in this research normalization was not the optimal technique as the biomarkers which were considered outliers were sometimes indicators that someone has liver disease or that they did not survive HCC. This method scales a feature to have a value between 0 and 1, where 0 is considered the minimum value in the feature and 1 is the maximum value in the feature.

The new value for each record in a feature is calculated using the formula:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

This method might be useful when all parameters need to have the same positive scale. However, the outliers from the dataset are lost.

(b) Standardization - this method was the recommended method when dealing with feature scaling and turned out to be the optimal method. This scaling technique transforms the values within a feature to have a mean of zero and a standard deviation of 1.

The new value for each record in a feature is calculated using the formula:

$$Z_i = \frac{x_i - \bar{x}}{s}$$

Here the mean feature value is subtracted from each particular record value in the feature and divided by the feature's standard deviation.

The most accurate model built using normalization was a model which also made use of the Random Forest Classifier, this model achieved F2 score of 73.48%. Meanwhile a model build using standardization, which used the k-nearest neighbour achieved F2 Score of 94.34%.

This is a F2 score increase of just under 21%.

The result of this scaling on the 'Total Bilirubin' and 'Ferritin' features can be clearly examined on the next page in Figure 3.4 and Figure 3.5.

In the histogram presented as Figure 3.4 we see that the ‘Total Bilirubin’ feature is squished very close together as its value ranged from 0 to 40.5 while the ‘Ferritin’ feature spans over the whole axis as its values range from 0 to 2230. Naturally the ML algorithms would view ‘Ferritin’ as a much more important feature.

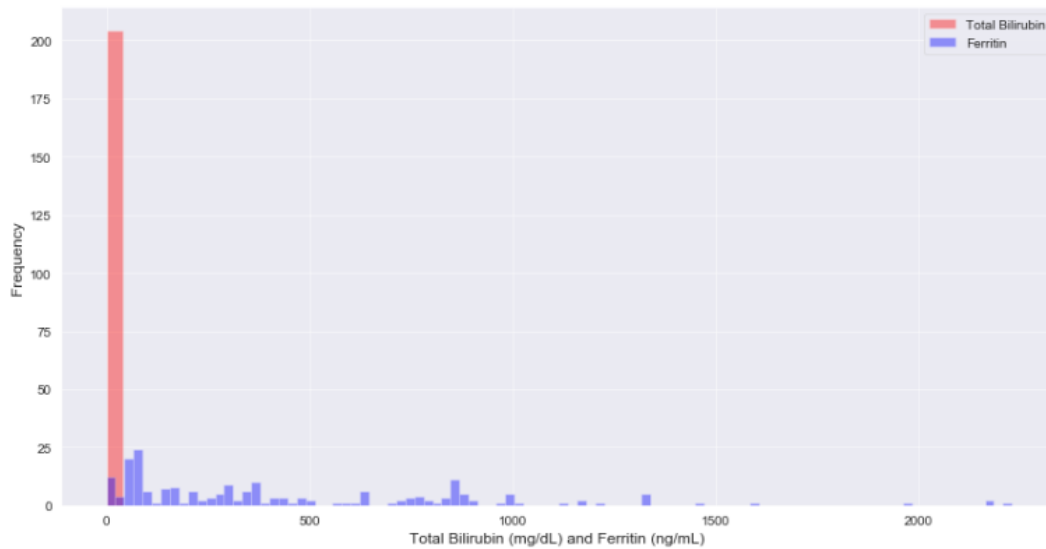


Figure 3.4: Features before Standardization.

After the features are standardized, as seen in Figure 3.5, that the two features overlap and are scaled properly, allowing the algorithms to view them as equally important.

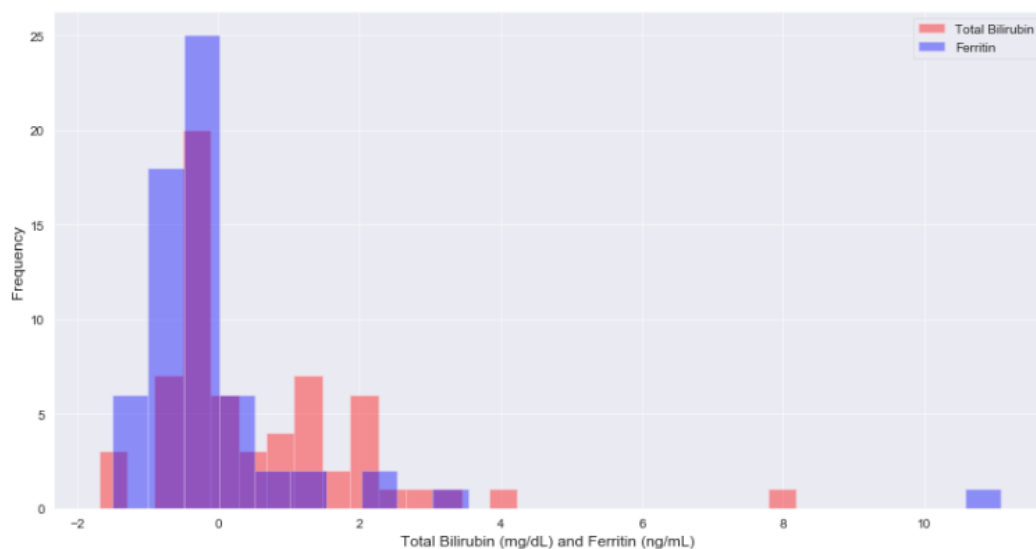


Figure 3.5: Features after Standardization.

3.3.3 Feature Selection

This step involves selecting which features the algorithm should have access to and learn from. This step is extremely important as any feature selected or disregarded may greatly increase or decrease the accuracy and F scores of the overall model. When building machine learning models one could potentially fall victim to making assumptions. For example, it is a common assumption that models built using more data should be more accurate than models built with less amount of data.

In fact, assumptions like these may cause a number problems:

(a) A vast number of features means that more computational work will need to be done in the training process. This combined with the fact that each model built has to then be evaluated using a 5-fold cross evaluation and that 5 different algorithms are used, means the overall training and testing time would be greatly increased.

(b) Some of the features selected may not be correlated with the target variable. This means that the feature has neither positive or negative impact on the target variable. For example, age and gender do not have any correlation as age does not impact gender and vice-versa, in contrast to this 'years of work experience' would have impact on 'income made by an individual' as one could assume that more experience results in a higher income and that higher income would mean someone has more experience. Including features with low correlation in a model creates noise and usually leads to overfitting (please refer to section 2.6.4 on page 35) which means the model will perform poorly with new data.

(c) Selecting a small number of features causes a problem opposite to the one presented in part (b). The model built with only a handful of features will be generalizing and therefore will have problems with underfitting. (please refer to section 2.6.4 on page 28)

This step may seem quite simple, one might think that just removing features with low correlation to the target variable and keeping the most correlated features will obtain the most accurate results.

However, this is not true as each machine learning algorithm may interpret the features selected differently.

This means that there is no optimal feature set and different sets of features will perform differently when different algorithms are used.

Please refer to appendix A to review the correlation heat maps the author generated and used to perform manual feature selection.

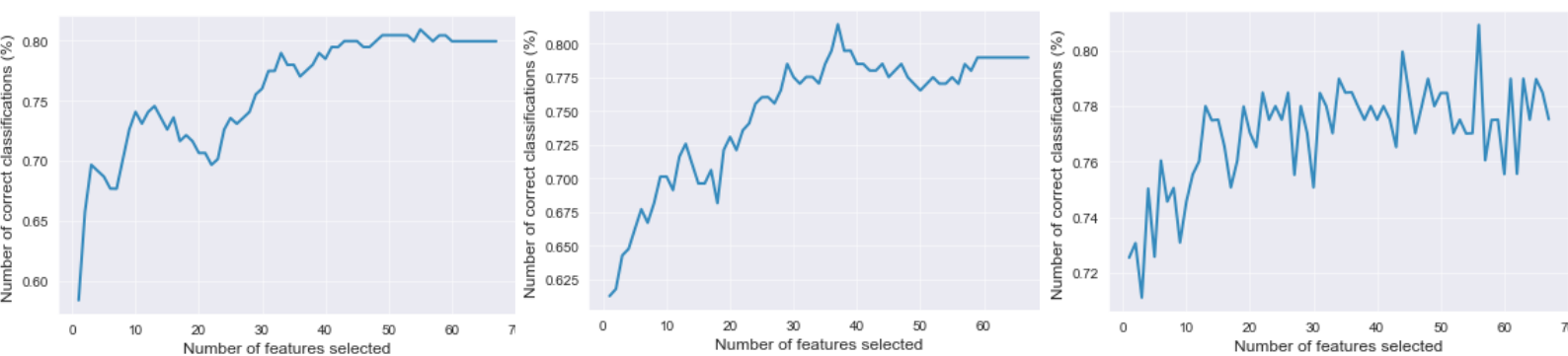
Therefore the author had to find a method to determine the optimal set of features for each algorithm used. This was achieved through a process called recursive feature elimination with cross-validation (RFECV). RFECV is a very demanding process, both in terms of time and computational power.

First the RFECV evaluates how a model would perform if all features were used, then it recursively eliminates the weakest feature, using these 3 steps:

1. Training and Testing - the model is trained and tested multiple times using cross validation with all available features. The accuracy of the model is recorded.
2. Least contribution - RFECV establishes which feature made the least contribution to the predictive accuracy.
3. Elimination - This feature is removed and the process repeats from step 1 until there are no features left.

This method did not work with the MLP Classifier and k-Nearest Neighbour Classifier. Due to the ambiguous nature of neural networks, the classifier can view different features as more or less important after every run hence it can not provide a definite set of features after RFECV. On the other hand k-Nearest Neighbour does not provide logic to do feature selection. This meant different method had to be implemented for these two classifiers. A simple “pen-and-paper” method was implemented. The model would be built with one less feature on every compile. If the predictive accuracy of the model decreased the missing feature would be deemed important and put back into the dataset. This process was repeated for all combinations of features.

The optimal number of features selected differed between algorithm, reinforcing the point that different algorithms make use of the available features to a different extent. These results can be seen in the figures below.



(a) Logistic Regression

(b) Support Vector Machine

(c) Random Forest Classifier

Figure 3.6: Accuracy to number of features selected for various algorithms (HCC Survival)

Figure 3.6 (a) presents that the accuracy of a model can significantly drop when features are added. The dip happens at around 22 features for this specific model.

In Figure 3.6 (b) however we notice that the accuracy of the model peaks at 38 features and drops as features are added to the model.

Figure 3.6 (c) presents how the accuracy of a model can dynamically change with each added feature and how much impact feature selection has on the overall predictive accuracy. Examples of similar behaviour regarding the Liver Disease dataset can be seen below:

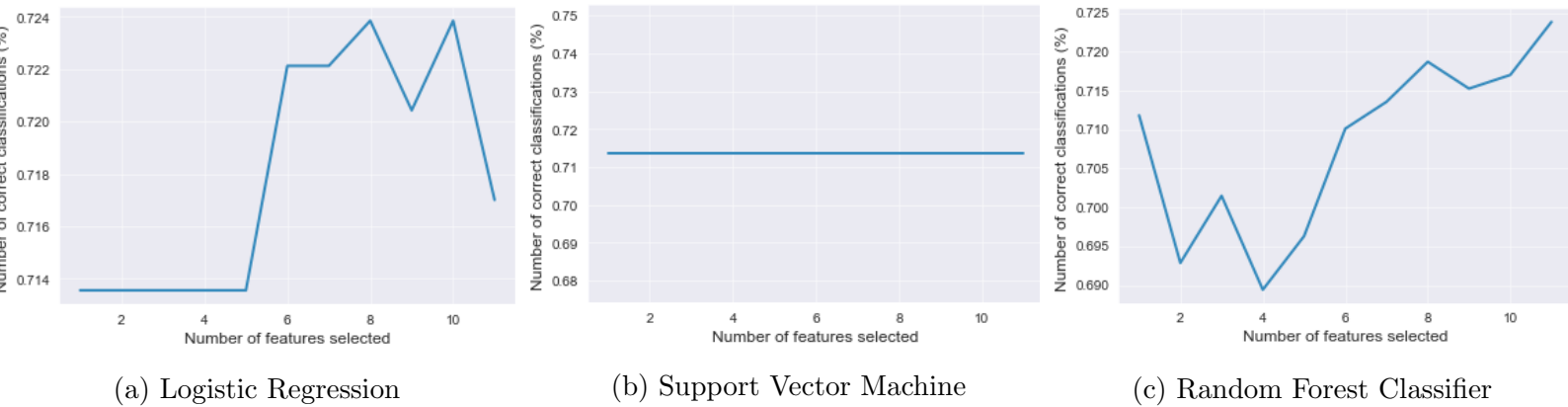


Figure 3.7: Accuracy to number of features selected for various algorithms (Liver Disease):

Figure 3.7 (b) is particularly interesting as it presents just a straight line, which would indicate that having one feature is just as good as having all the features. The explanation for this is actually within the name of the algorithm. SVM uses support vectors and in this specific example SVM deemed only one of the features important enough to separate the data. Although additional features were added SVM only used “alanine transaminase” for the support vectors. Therefore, as the features were added one by one the hyperplane separating the data points didn’t change. Please refer to appendix C to see the feature importance for the SVM algorithm.

These findings present that some features may be seen as more or less important by different machine learning algorithms. The scikit-learn library provides a visualization option for feature importance, however this feature is only available for tree based algorithms. The ranking of feature importance can be seen below in Figures 3.8 and in appendix C. Please note that features which were not important have been dropped and therefore are not ranked.

Different algorithms selecting different features and then ranking these features based on importance was an extremely interesting observation. Seeing algorithms such as SVM learning more progressively as features are added, in contrast to Logistic Regression which learned quickly from small number of features was also quite intriguing. Of course the objective of the research was to maximise the F1 and F2 scores of the models whilst keeping a high enough accuracy, with no regard for time or resources needed to train the models.

The result of this extensive process of feature selection can be viewed in the ‘Algorithm Comparison’ section starting on page 49, section 4.2.

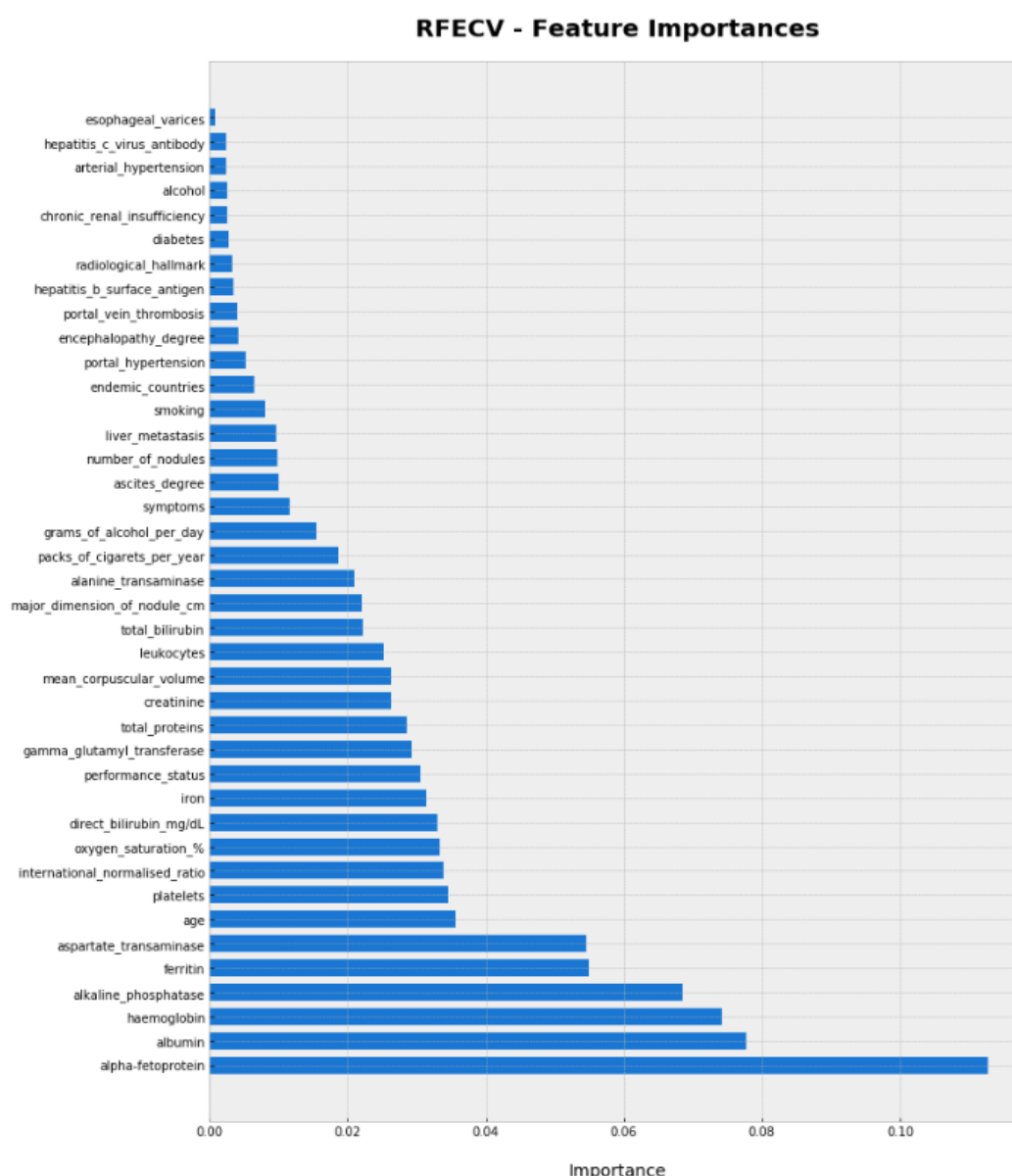


Figure 3.8: Ranking the importance of different feature visualized (HCC Survival).

3.4 Model Training

Once the data was correctly processed to ensure the highest possible F1 and F2 scores could be achieved by each individual algorithm the model training process could begin. The results of this process are compared and contrasted in Chapter 4, section 4.1. Model training has multiple stages just the way data pre-processing does. Once again, different methods and techniques had to be implemented to further increase the F scores of models. This section will focus on hyperparameter tuning and cross-validation.

3.4.1 Hyperparameter Tuning

Hyperparameters are parameters which can be passed into each machine learning algorithm prior to the training process. Usually hyperparameters are chosen based on the characteristics of the data.

The technique used for hyperparameter selection in this research was called Exhaustive Grid Search (EGS). This method involves trying every possible combination of hyperparameter options and evaluating each combination until the most optimal set is established. This process is immensely time and resource consuming.

For example, if 4 hyperparameters are available, each with 5 distinct options then 1024 models have to be built and compared.

If each model had to be evaluated using 5-fold cross-evaluation this number would rise to 5120 models built in total. Assuming one build takes 30 seconds, it would take almost 43 hours of non-stop model building to evaluate all possible options for just one algorithm.

On top of that, all the models built are stored in memory. The computer used for this research ran out of memory on couple of occasions and was unexpectedly force restarted. This meant only the hyperparameters which the author suspected could potentially increase the predictive accuracy were evaluated. Of course all algorithms have multiple hyperparameters and learning about each one significantly increased the time taken for this process.

A table of the hyper-parameters chosen for each algorithm can be seen on the next page along with the increase in predictive accuracy the hyperparameter tuning process achieved. The “accuracy increase” displayed stands for the increase in accuracy acquired thanks to hyperparameter tuning as opposed to the accuracy which would be achieved if all hyperparameter values were left as default. (If a hyperparameter does not appear it means the default value was used).

Table 3.2: Predictive accuracy increase achieved through Hyperparameter Tuning. (HCC Survival)

Algorithm	Hyperparameters	Highest Accuracy Achieved	Accuracy Increase
MLP	MLPClassifier(alpha=0.01, hidden_layer_sizes=(50, 100, 50), learning_rate='invscaling', max_fun=15000, max_iter=500)	83%	18%
SVM	SVC(C=100, degree=5, gamma='auto', kernel='poly')	76%	3%
k-NN	KNeighborsClassifier(leaf_size=10, n_neighbors=1, p=1, weights='uniform')	84%	10%
Random Forest	RandomForestClassifier(bootstrap=False, criterion='entropy', min_samples_leaf=3, min_samples_split=3, warm_start=True)	82%	7%
Logistic Regression	LogisticRegression(C=0.1, max_iter=300, multi_class='ovr', solver='liblinear', tol=0.0001)	80%	4%

3.4.2 Cross-Validation

For every model built the dataset being used has to be split into a training and a testing set. The training set is used to teach the model while the testing set is used to evaluate the accuracy and other metrics of the model. Multiple splits were used over the course of this research, these were: 80/20%, 70/30%, 60/40%, the recommended 70/30% proved to be most advantageous.

Please keep in mind that the two datasets used are quite small, which may lead to few problems. When data is split the model can not learn all the information the dataset possesses. This means that the testing set contains information which could be vital to the training process as it could completely change the reasoning of the model. On top of that the training set could contain a lot of data which does not add much new information to the model hence, teaching it certain things over and over and not letting it generalize, which may lead to underfitting. (Please refer to section 2.6.4.)

To ensure that all models built are trained and tested with the greatest amount of data k -fold cross-validation is used. This technique means once again splitting the dataset into k number of folds.

Each model is then trained k times, and tested against different, unique fold each time. The mean of the results of each fold is found and displayed. Five was selected as the optimal number for k . This number was selected based on prior literature review of other, similar studies.

3.5 Summary

In this chapter the author focused on the implementation process conducted over the course of this research. The author explained his reasoning behind the technologies selected and described the way in which the data was collected. The author then went on to explain all the data pre-processing techniques and model training techniques which he considered and tested. The author also gave a detailed account on why some of the techniques were more suitable better than others.

Chapter 4

Evaluation

After data pre-processing and model training the research reached its final stage, which was evaluation. Of course, the algorithms and methods used were compared, contrasted and evaluated at almost every stage of the model building pipeline. This chapter deals with the final evaluation of the most accurate models and methods used to build them, for both the liver disease diagnosis and hepatocellular carcinoma survival prediction datasets. The models and methods used were evaluated using a range of metrics, which are explained next, in this chapter.

4.1 Evaluation Metrics

Each model evaluation was based on predictive accuracy, precision and recall, all in the form of percentages. These terms are explained below. Tables with the exact numerical results are present in the section 4.2. Please refer to the table 4.1 below to get familiar with some of the vocabulary used in this chapter.

Table 4.1: Types of classification made by models.

Type of Classification	Explanation
True Positive (TP)	Model <i>correctly</i> predicted a <i>positive</i> class
True Negative (TN)	Model <i>correctly</i> predicted a <i>negative</i> class
False Positive (FP)	Model <i>incorrectly</i> predicted a <i>positive</i> class
False Negative (FN)	Model <i>incorrectly</i> predicted a <i>negative</i> class

4.1.1 Predictive accuracy

The first metric used to compare each algorithm was simply the predictive accuracy measurement. Predictive accuracy is the most popular metric used to evaluate machine learning classification models. This metric is extremely simple to implement, easy to understand and in most cases provides quite good feedback on the performance of the model. However, accuracy is such a general metric that it may not always be adequate. The formula for it is:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad \text{or} \quad Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

This metric may become inaccurate and wrong to use when dealing with unbalanced datasets and especially unbalanced medical datasets.

For example, if a dataset of 1000 patients was taken, where 999 of the patient are cancer free and 1 patient has HCC, the model could predict that all of the patients are cancer free and be 99.9% accurate.

The problem is that although the model seems accurate, one of the patient would never be diagnosed.

In this case the positive class is hugely outnumbered by the negative class and the model focuses on the negative class when it should focus on identifying the positive cases.

4.1.2 Recall

Recall was the second metric used. Here the evaluation focuses on finding all the relevant cases within a dataset. The formula for recall is:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

The issue with recall can be explained using the same scenario as illustrated in section 4.1.1. Here however, the model could predict that all the patients have HCC and get a recall score of 100%, indicating a perfect classifier has been found. Of course, this would not be true as perfect classifiers do not exist, especially in the medical field as human physiology is never 100% predictable due to its complexity.

4.1.3 Precision

In contrast to recall, precision is the number of positive class predictions that really belong to the positive class. In other words, it is the model's ability to identify only the relevant data points.

The formula for precision is:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Precision and recall work together, to provide more information about the model's predictive capabilities. In the example mentioned above, when recall would be 1.0, precision would be 0.0 indicating that there is a problem with the model. If the dataset used was more balanced and the model would correctly classify 1 patient with HCC, the model would achieve precision of 1.0 (as there was no false positives found) but a recall of 0.0. Thus, as we increase precision we decrease recall and vice-versa.

4.1.4 F Score

The first model which labeled all patients as cancer free was not useful as it would mean that 1 patient could potentially die simply due to using the wrong metric. That model had almost perfect accuracy, but 0 precision and 0 recall because it did not find any true positives. The second model classified everyone as cancer victims and achieved recall of 1.0 and precision of 0.0, meaning everyone would have to be tested further, making the model useless. The F score takes both metrics into account, using the following formula:

$$F_{\beta} = (1 + \beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

If a balanced classifier with the perfect balance of recall and precision is sought after, then the β parameter would be set to 1.

The β parameter is changed depending on what one considers more important in a given problem. For example F2-Score would put more emphasis on recall than precision. This makes the F2 score more suitable as the metric used for applications such as disease diagnosis or survival prediction as in this case, it is more important to classify correctly as many positive samples as possible, rather than maximizing the number of correct classifications.

4.2 Algorithm Comparison

Each model evaluation was based on predictive accuracy, precision, recall and the F-Score. 2 was used as the β parameter for the F-score as the author felt both the models sought after should be more biased towards finding all the true positives rather just trying to classify everything correctly.

In the author's opinion it is much more beneficial to tell a patient they have a disease and later find out they really do not after more testing, than to tell a patient they do not have a disease and later find out they do, when it could be too late to take action.

The results for the F1 score obtained for HCC survival prediction and Liver Disease diagnosis can be seen in Figures 4.1 and 4.2 respectively.

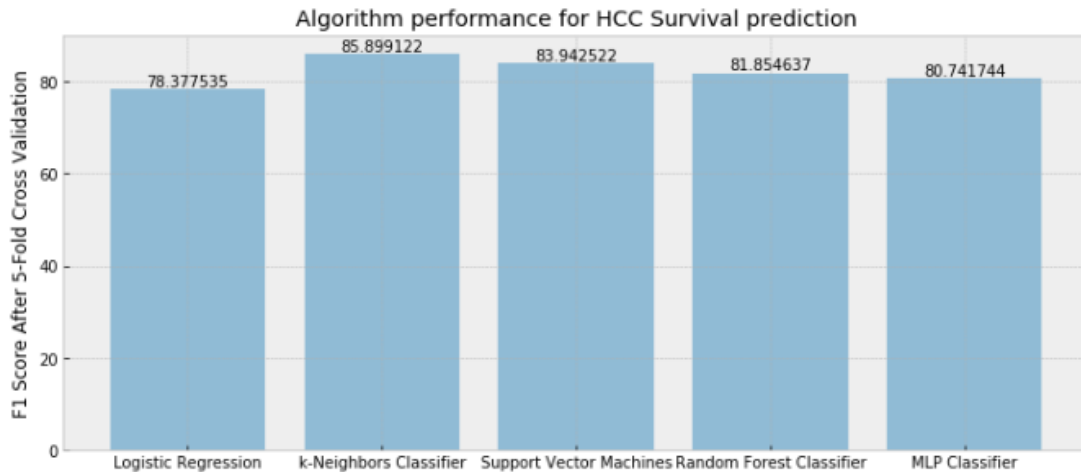


Figure 4.1: F1 Score obtained by all algorithms after 5-fold cross validation (HCC Survival).

The k-Neighbours Classifier performed the best out of all the algorithms achieving F1 of 85.9% and was closely followed by the SVM and Random Forest Classifiers which got F1 of 83.94% and 81.85% respectively. The author was not surprised with the results.

The k parameter was set to 1 meaning only one neighbour was used for the classification. Although this may seem like a bad practice, this parameter achieved the highest accuracy, F1 and F2 score.

This is because the patients who were used for classification almost always had very elevated biometrics such as $\alpha - fetoprotein$.

For example, when the patient being classified had elevated $\alpha - fetoprotein$ and was found to be close to a deceased patient who also had elevated $\alpha - fetoprotein$, the algorithm knew the first patient will also die.

Using 1 as the k parameter greatly reduced noise, meaning the algorithm could make concrete decisions.

The problem with the MLP classifier was the fact that neural networks usually perform better on much bigger datasets. In this case the model was basically too sophisticated for the tiny dataset being used.

However, it performed slightly better than the author expected.

Although some of the data seemed like it could be separated easily using logistic regression, this assumption was quite wrong. Once again, the vast amount of features became problematic.

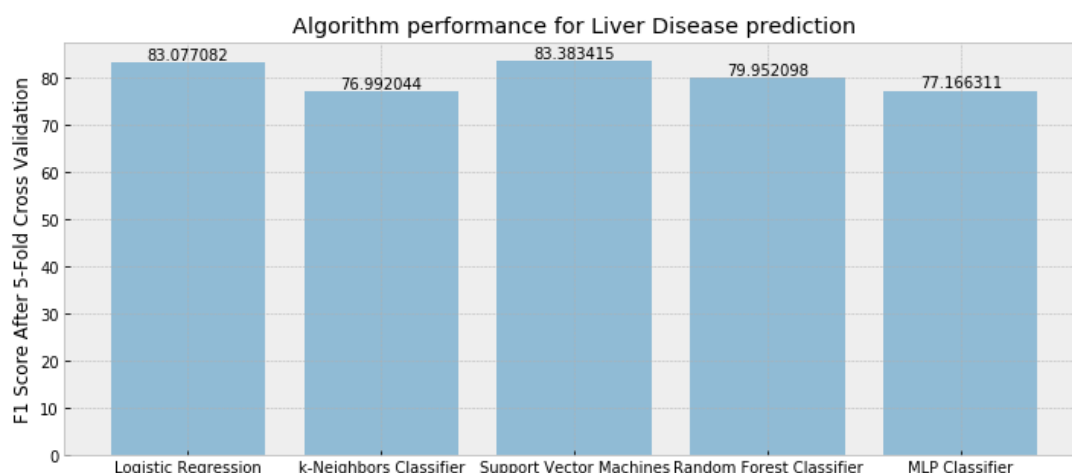


Figure 4.2: F1 Score obtained by all algorithms after 5-fold cross validation (Liver Disease).

In Figure 4.2 we see almost completely different results. SVM performed the best achieving F1 score of 83.38%, followed closely by logistic regression which got 83.07%. This can be explained by the fact that the dataset used for building this model had many records but only 10 features (not counting the target variable). The SVM model and Logistic Regression model had a much easier job reasoning and separating the data. Once again, there was simply not enough data for the MLP classifier to excel.

The results achieved by the Random Forest classifier were quite mediocre, for both of the datasets, possibly because it is prone to bias when creating each individual model, as each tree created uses only a subset of the available data. The fact that k-Neighbours performed significantly worse was quite surprising. The author concluded this area required more investigation.

As said before, the author considered high F2 score to be a more optimal evaluation metric for medical classification. High recall would mean a larger number of false positives, however in case of both disease prediction and survival prediction it is safer to have more false positives than false negatives. The patients classified as false positives would eventually find out they are not sick, whereas the patients classified as false negatives could never find out they are sick.

The results for the F2 score obtained for HCC survival prediction and Liver Disease diagnosis can be seen in Figures 4.3 and 4.4 respectively. These results are extremely interesting as it is clearly seen that the algorithms which performed well using the F1 score, performed even better when the F2 score was used as the metric, whereas algorithms which performed poorly or in the medium range performed worse.

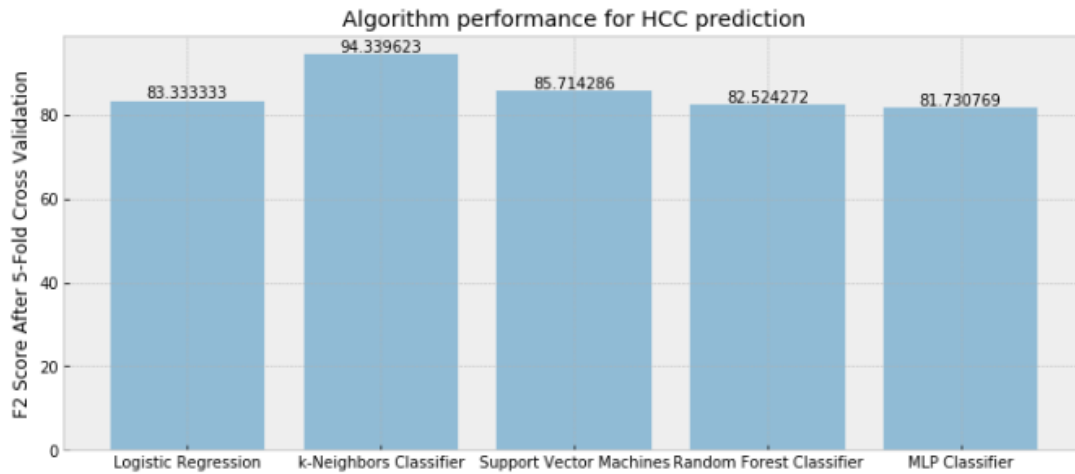


Figure 4.3: F2 Score obtained by all algorithms after 5-fold cross validation (HCC Survival).

The results seen in Figure 4.4 are particularly interesting as the F2 score for the SVM model was almost 10% higher than F1 Score. This means that the SVM was much better at finding all the true positives and yet optimal enough to not have a low precision and therefore not classify many patients patients in a way that would lead to many false positives. This made SVM almost an ideal classifier which could potentially be used by practitioners if it was further perfected.

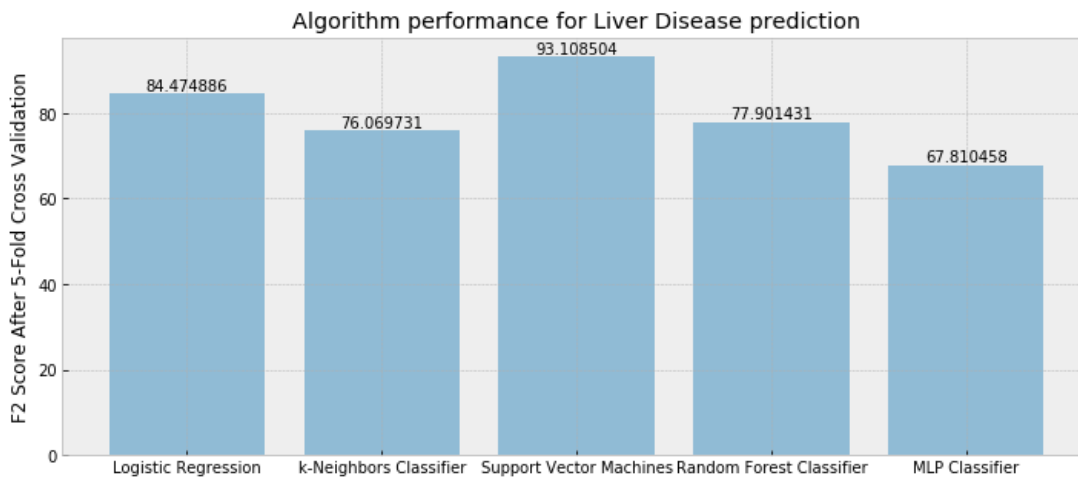


Figure 4.4: F2 Score obtained by all algorithms after 5-fold cross validation (Liver Disease).

As the author mentioned before, accuracy should not really be the metric used for evaluation in this area. However, please refer to appendix B for accuracy related results.

4.3 Importance of Data Pre-Processing Techniques

One of the objectives of the research was to investigate what impact data-preprocessing techniques would have the different models built. The author decided to investigate using all the algorithms selected. The results are presented in Figure 4.5 below.

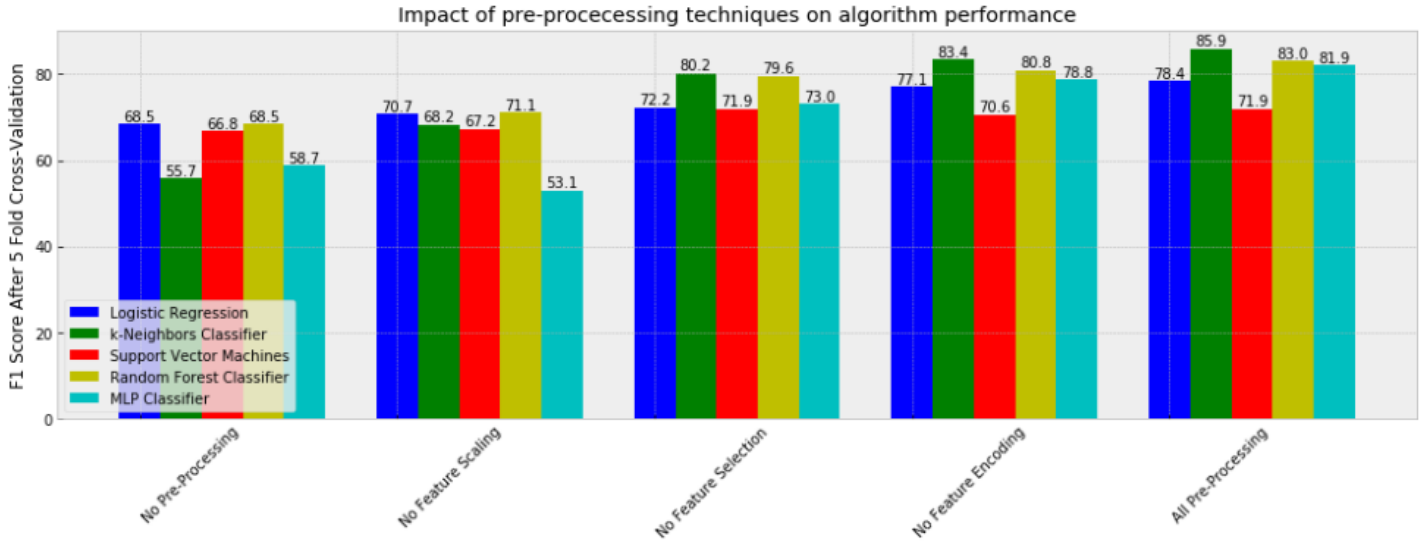


Figure 4.5: Impact of data pre-processing on the F1 score of various algorithms (HCC Survival).

It is clear that every stage in data pre-processing had a positive impact on the F1 score of all algorithms. It is also quite clear that different stages impacted different algorithms to a varying degree. For example, the F1 score of k-Nearest Neighbour increased by a total of 30.2%.

It is worth noting that k-Nearest Neighbour performed the worst before data pre-processing and the best after data pre-processing.

Feature encoding did not have a huge impact on the F1 score, which is understandable as only 3 features were one hot label encoded.

In addition to that, some of these features were later dropped when feature selection was performed, and only appeared in some of the “optimal feature sets”. The impact of feature scaling is very interesting as it seemed to impact the MLP classifier and k-NN classifier very noticeably, but increased the F1 score of logistic regression very slightly. The MLP classifier is sensitive to scaling as inputs are intertwined in the back-propagation process. Feature selection had surprisingly very little impact, this is because although some of the features had very low correlation to the target variable they did not impact the classifiers negatively. In other words, some features had very high positive impact on F scores and accuracy and some had virtually no impact, neither positive nor negative, hence it didn’t matter if they were kept or dropped.

The process of feature selection was very important however as it presented the author with features which are important and had to be kept in almost all “optimal feature sets” for all models.

4.4 Importance of Training/Testing Techniques

This chapter ends with the evaluation of model training and testing techniques and specifically hyperparameter tuning and cross-validation. These two processes focus only on the models and not on data itself. Hyper-parameter tuning had significant impact on all algorithms used which can be seen in table 3.10. The number of hidden layers and the sizes of them along with the number of max iterations allowed had the greatest impact on the MLP classifier. This is probably due to the fact that the HCC dataset had a great number of features and the MLP classifier needed an increase in both of these parameters to really learn all the information it was provided with efficiently.

Out of all the processes performed over the course of the research, on average, hyper-parameter tuning and feature scaling had the greatest positive impact on the accuracy and F Scores across all the models built. This fact, reinforced what Sato, M. (2019) noted in his study of HCC diagnosis, hyperparameter tuning is extremely important and should never be overlooked, no matter how time and resource consuming it is.

The last method used by the author to validate his results was 5-fold cross validation. This technique is different from all the other used in this research as it did not intend to improve the accuracy or F scores of the models. Cross validation is performed to increase the integrity of the results used for evaluation. As mentioned in section 3.4.2, splitting data into training and testing sets can become dangerous as it may deny the model some crucial learning experiences. Some of the records included in the testing set could have a huge impact on the evaluation metrics. Cross-validation insures that all data is used for both testing and training, providing more complete results.

The results of this process can be seen below in Figure 4.6 below:

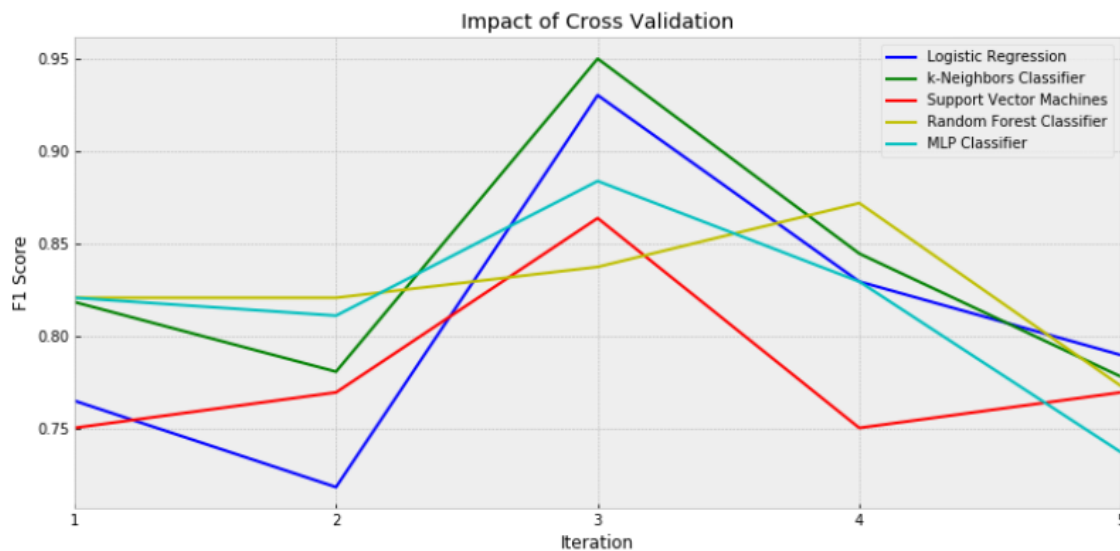


Figure 4.6: 5-Fold Cross Validation performed on the HCC Survival Dataset.

Please note the significant amount of variation between the F1 Score of every single model. For example the model built with the subset of the dataset used on iteration 3 achieved a much higher F1 Score than subset used on iteration two. However, this does not mean that the model itself was more accurate, it only means that the model was trained and tested on different subsets of the data which may have been beneficial for some of the algorithms, yet degrading to others.

All the experiments performed and presented in this chapter made it clear that all the techniques used in both data pre-processing and model training were mostly successful to varying degrees. All the models and algorithms were evaluated with a multitude of metrics, which in turn demonstrated that the k-Nearest Neighbour classifier was the most suitable for HCC survival prediction and that the SVM classifier was the most suitable for liver disease diagnosis.

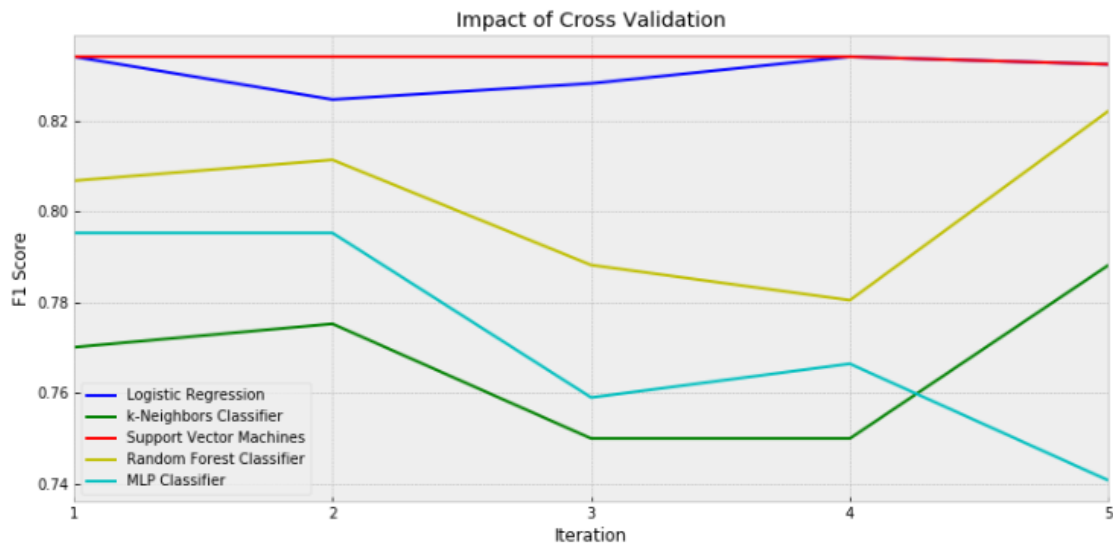


Figure 4.7: 5-Fold Cross Validation performed on the Liver Disease Dataset.

Chapter 5

Conclusion

The objective of this research was to determine if machine learning could be used for prediction of liver disease and hepatocellular carcinoma survival based on patients' biomarkers. This objective could only be considered as completed once the predictive accuracy, F1 and F2 score achieved by the models built was regarded as very high.

Chapters two to four give a detailed description of the work done to answer the research question and achieve the objectives set by the author.

1. Chapter 2 consisted of extensive research into: machine learning, machine learning techniques, medical data science, liver disease and cancer research. Multiple studies have been reviewed and contrasted, which allowed the author to extract the most useful information and learn quickly. The author explored and studied multiple machine learning algorithms, their pros and cons, complexity and usefulness in terms of classification of various medical datasets.

Although the research was not a novel idea, it built on prior studies and even achieved accuracy higher than some of them.

Comparing the results achieved by Nandipati S., Shamsudin H., et al. (2019) to the results achieved by the author of this research (please refer to sections 2.4, 4.2 and also to appendix B) it is quite evident that the author of this research achieved much better result than his predecessors. In fact, every algorithm the author chose outperformed the algorithms in the prior study, with SVM being the only exception.

The increase in accuracy ranged between 4% and 8%, which can be seen as quite high. The highest accuracy Nandipati S. and Shamsudin H. achieved was 81.81%, whereas the author of this research achieved predictive accuracy of 92.68%. On top of that, the author achieved higher recall, which could be seen as the more important metric.

By examining and outlining previous research in this area, the author achieved the first objective of the research presented in section 1.3.

2. The author was not presented with any data prior to the research, therefore data had to be collected. The author searched all open source data repositories and even tried to contact multiple other researchers who performed similar studies, to obtain the data they have used when conducting their studies. Ultimately two datasets which allowed this research to happen, were found and used, thus the second research objective was achieved.

3. The third research objective involved learning and using different techniques for:

- a) Cleaning the data. b) Missing Data. c) Data Formatting.
- d) Outlier detection. e) Feature Engineering. f) Feature Encoding.
- g) Feature Scaling. h) Feature selection. i) Hyper-parameter Tuning.
- j) Cross-Validation. k) Algorithm selection and comparison.

All of the stages above have been carefully explored and adapted to ensure the highest possible predictive accuracy, F1 and F2 scores have been achieved. Therefore the third research objective was completed successfully.

4. In Chapter 4 the author evaluated and presented the F1 and F2 score of each algorithm chosen. The models were carefully studied and optimized to the author's full ability. From the results it can be concluded that machine learning models can in fact be used as devices for liver disease diagnosis and for hepatocellular carcinoma survival predictors. Of course the diagnosis could not be performed by the models alone and medical practitioners would still need to review patients and the models themselves on regular basis.

Please also note that the research has been done with a very limited amount of data and the performance of the models could only be truly evaluated with much bigger datasets. Taking everything into consideration, the author feels that the research question has undoubtedly been answered.

Yes, machine learning models can, and will be used to diagnose liver disease and predict patients survivability. However this can only be achieved through a very detailed, extensive and exhaustive research into all stages mentioned throughout this project, all the way from literature review through data collection and pre-processing ending at hyperparameter tuning, cross validation and evaluation.

k-Nearest Neighbour algorithm performed the best when it came to HCC survival prediction achieving F1 score of 85.89% and was closely followed MLP Classifier and Random Forest Classifier which achieved 81.94% and 81.63% respectively. These accuracies, although they are quite high, are still not high enough for medical practitioners who require extremely precise results. However, these results can be considered quite excellent when the time frame given for this research is considered.

In terms of Liver Disease diagnosis k -Nearest Neighbour algorithm performed the worst achieving F1 score of only 76.99% with SVM and Logistic Regression both beating it with 83.38% and 83.08% respectively. When evaluating the results it becomes quite clear that the algorithms chosen were quite appropriate for the task at hand. In fact, at different stages in the data pre-processing and model training every algorithm was considered ‘the best’ at some point. This leads to a conclusion that every algorithm could in fact achieve a better result than it did, if more time was spent on hyper-parameter tuning, and data manipulation to suit that specific algorithm. The results and observations could potentially guide other researcher who will work in this specific area, exposing pitfalls while presenting promising approaches and methods.

It is also very important to mention that although the results achieved from this research may not be of high enough accuracy for medical practitioners it is extremely important that work in this field continuous as machine learning could one day be the answer to most, if not all, of our medical problems.

5.1 Future Work

It is very important to recognize that the project was not perfect and had many flaws and faced challenges. However, these flaws and challenges are a great indicator for what can and should be improved in the future.

This section provides a description on where the project could be improved and in what direction it could be taken. This section is also crucial for any researchers who might review this study as it potentially provides them with a clear stepping stone for their own research. The author established some areas where further investigation could possibly increase the predictive accuracy and the F2 Score of the models built.

These areas are listed below:

1. Data collection - The datasets used for in this study were very small and unbalanced. The research would definitely benefit from much bigger, more balanced datasets which, as mentioned before, do exist. Of course this problem has to respect the ‘quality over quantity’ cliché. Unless complete data with features that matter to the classification we are trying to achieve is collected, the models built will never be perfect or of high enough accuracy. On top of that, models built using datasets acquired from just one ethnic group, will only be able to correctly classify people within that ethnic group. This is because the biomarkers mentioned throughout the research vary between these ethnic groups; for example, people from South Asia suffer from non-alcoholic fatty liver disease (NAFLD) more than any other group of people in the world.

To achieve better results the problem could be divided into many sub-problems, which would involve building models for each ethnic group, or considered as one big problem where a much more complete model would be built with a huge dataset made of all the available records.

2. Oversampling - The datasets in this research were balanced using the Synthetic Minority Over-sampling Technique. Of course creating synthetic records is not ideal for any machine learning application. Having said that, there are many ways to perform such task and SMOTE is only one of them. In fact, there are even multiple ways to implement SMOTE.

Of course, each of these ways would have to be explored and correctly implemented to balance the datasets, in a way that the synthetic patients would mimic real patients as accurately as possible.

However, this step would not be necessary if data collection was performed as described in step 1.

3. Algorithm selection - There are at least 75 machine learning algorithms known to the author. This research only utilizes 5 of these.

Of course there exists a possibility that an algorithm which has not been used in this research would perform much better on the datasets used. Of course every algorithm requires different data manipulation techniques prior to model training.

Testing all the algorithms would take an incredibly long time, especially if exhaustive hyper-parameter tuning and cross-validation were performed.

However, this is definitely an area worth exploring in the future.

4. Missing Values - The author used several imputation methods for filling in missing values: mean, median, mode and k-NN imputation.

However neither of these is perfect as they all impute the missing values using information which has already been previously in the dataset.

This method reinforces the way the model thinks rather than broadening its capabilities. Other methods which should be explored include:

(a) Missing Indicator Value - here the missing value is replaced with an indicator such as “-1” which tells the algorithm to skip over this value when learning. This method doesn’t provide any new information, however it also doesn’t force the model to reason just a certain way.

(b) Deleting Rows - this methods was not used as there were many missing values present in the datasets, dropping all the rows would mean that almost 80% of the data would be lost. However, this method could be used on a much larger dataset or if the dataset didn’t have many values missing.

5. Data Scaling - Data can be scaled in many different ways. The author used normalization and standardization and concluded standardization was the better option for the problem at hand. Yet, there are many unexplored scaling methods such as: binarizing, robust scaling, or using a normalizer (different to normalization as it is row based rather than column based). As said before different algorithms could favour different scaling techniques, therefore this is another area worth exploring.
6. Feature selection - Once again, different algorithms favour and learn from various features differently. Performing exhaustive search for the perfect set of features for each algorithm should definitely be repeated and reviewed by a professional data scientist and possibly a medical practitioner. When dealing with data, and especially medical data, one should be extremely careful of what data they drop as any given feature may prove vital to the success of the study.
7. Hyperparameter tuning - As mentioned in section 3.4.1, finding the optimal hyperparameters for just one algorithm can be a very time and resource consuming process, as it can only be done right with the Exhaustive Grid Search method which tests every possible combination of hyperparameters. Doing so may result in models being built and evaluated for days if not weeks. However, if doing so would increase the accuracy of the models up to a medically approved standard, this work has to be performed.
8. Medical and Data Science Knowledge - The author possessed a very limited amount of medical and data science knowledge prior to this study. If more knowledge about: medical data, biomarkers, liver disease, liver cancer, data science and machine learning algorithms was acquired the author could approach this study again much more efficiently which could potentially mean much better results.

5.2 Final Remarks

The author faced many challenges, some common to machine learning (please refer to section 2.6) some common to research itself. Some of the main challenges are listed below.

Lack of data - As said before, lack of data was a huge challenge. The author did not receive the dataset he was originally meant to work with due to legal reasons. The data collection process was therefore very stressful as the project could not go ahead without a suitable dataset.

Time Management - Although time management was a challenge it was also one of the main reasons the project became a success. The author quickly learned that there is no perfect path or approach to take.

The only way to truly achieve optimal results was through trial and error. This approach became very frustrating at times, as there are just so many different methods for implementing each stage of the machine learning pipeline, that finding the ultimate combination is difficult.

Of course, even if the best possible combination was found the author wouldn't know and would continue to implement different methods which could potentially lead nowhere.

However, the author managed to split the time allocated for the project quite well between literature review, implementation, evaluation and project write up and stayed vigilant in his time keeping.

Bugs - This challenge is closely related to time management. Building machine learning models takes time. The shortest build time over the course of this research could be calculated in milliseconds, the longest took just over 25 minutes. Bugs in the implementation, be it in the data pre-processing or visualization stages mean that the whole training process has to be repeated. This challenge made the author much more aware and careful about any changes to the code used. The author simply could not afford to make any errors due to the time constraint.

If the author was to start the project anew all the ideas mentioned in section 5.1 would have to be implemented or at the very least reviewed.

All the knowledge about machine learning and data science acquired over the course of this research would be extremely useful if learned prior to this project. The author came to such a conclusion as, a lot of the time allocated for the project was spent learning about ML techniques rather than implementing them.

Overall, the author considers this research as a success as he was keen to improve his computer science and data science knowledge and skills.

Of course this has been achieved as the project was challenging, making the author learn, evaluate and reflect on day to day basis, greatly increasing his skills and skill-set.

The author also learned so much about python, various python libraries and other useful tools, all required by someone who wants to pursue a career in data sciences.

Bibliography

- [1] Russell, S. and Norvig, P. (2010).
Artificial Intelligence: A Modern Approach. 3rd ed.
- [2] Mitchell, T., (1997).
Machine Learning. 1st ed. New York, NY, USA: McGrawHill, Inc. isbn: 0070428077.
- [3] Science Museum, n.d. How Do Healthy Cells Become Cancerous?
<http://whoami.sciencemuseum.org.uk/whoami/findoutmore/yourbody/whatisncancer/whathappensincancer/howdohealthycellsbecomecancerous>
- [4] Gu, J., (2013). *Primary Liver Cancer*, Dordrecht: Springer, pp.399-400.
- [5] Mintz, L., (2020). World Cancer Day: What Is The Most Common Cancer In The UK, And Which Has The Worst Survival Rate?
The Telegraph.
<https://www.telegraph.co.uk/health-fitness/body/world-cancer-day-common-cancer-uk-has-worst-survival-rate/>
- [6] Pinter M., Trauner M., Peck-Radosavljevic M., Sieghart W. (2016).
Cancer and liver cirrhosis: implications on prognosis and management.
ESMO Open. 2016;1(2):e000042. doi: 10.1136/esmoopen-2016-000042.
eCollection 2016. Review. PubMed PMID: 27843598; PubMed Central PMCID: PMC5070280
- [7] Mayoclinic.org. (2019) Hepatocellular Carcinoma - Overview - Mayo Clinic.
<https://www.mayoclinic.org/diseases-conditions/hepatocellular-carcinoma/cdc-20354552>
- [8] Dasu, T. and Johnson, T., (2003).
Exploratory Data Mining And Data Cleaning. Hoboken, NJ: John Wiley et Son.
- [9] Ayer T., Alagoz O., Chhatwal J., Shavlik JW., Kahn CE. Jr., Burnside ES. (2010).
Breast cancer risk estimation with artificial neural networks revisited: discrimination and calibration.
Cancer. 2010 Jul 15;116(14):3310-21. doi: 10.1002/cncr.25081. PubMed PMID: 20564067; PubMed Central PMCID: PMC2920215.
- [10] Cruz JA., Wishart DS.. (2007).
Applications of machine learning in cancer prediction and prognosis.
Cancer Inform. 2007 Feb 11;2:59-77. PMID: 19458758; PMCID: PMC2675494.

- [11] Sayed, S., (2018). *Machine Learning Is The Future Of Cancer Prediction*.
<https://towardsdatascience.com/machine-learning-is-the-future-of-cancer-prediction-e4d28e7e6dfa>
- [12] Kouroua, K., P. Exarchos, T., P.Exarchos, K., V.Karamouzis, M. and I.Fotiadis, D., (2014). *Machine Learning Applications In Cancer Prognosis And Prediction*.
<https://www.sciencedirect.com/science/article/pii/S2001037014000464>
- [13] Montazeri M., Montazeri M., Montazeri M., Beigzadeh A. (2016). *Machine learning models in breast cancer survival prediction*.
 Technol Health Care. 2016;24(1):31-42. doi: 10.3233/THC-151071.
 PubMed PMID: 26409558.
- [14] Xie X., Hu Y., Jing Ch., et al. (2017)
A Comprehensive Model for Predicting Recurrence and Survival in Cases of Chinese Postoperative Invasive Breast Cancer.
 Asian Pac J Cancer Prev. 2017;18(3):727–733. Published 2017 Mar 1.
 doi:10.22034/APJCP.2017.18.3.727
- [15] Sato, M., Morimoto, K., Kajihara, S., Tateishi, R., Shiina, S., Koike, K. and Yatomi, Y., (2019)
Machine-Learning Approach For The Development Of A Novel Predictive Model For The Diagnosis Of Hepatocellular Carcinoma.
<https://www.nature.com/articles/s41598-019-44022-8#citeas>
- [16] Nandipati S., Shamsudin H., XinYing C., (2019)
Classification and Feature Selection Approaches by Machine Learning Techniques: Hepatocellular Carcinoma (HCC) Prognosis Prediction.
 Amity Journal of Computational Sciences (AJCS), Volume 3 Issue 1, ISSN: 2456-6616 (Online)
- [17] Pupale R. (2018).
Support Vector Machines (SVM) - An Overview.
<https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>
- [18] Bambrick, N., n.d.
Support Vector Machines: A Simple Explanation - Kdnuggets.
<https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html>
- [19] Navlani, A., (2018).
KNN Classification Using Scikit-Learn. DataCamp Community.
<https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>

- [20] Medium. n.d.
Applying Random Forest (Classification) — Machine Learning Algorithm From Scratch With Real.
<https://medium.com/@ar.ingenious/applying-random-forest-classification-machine-learning-algorithm-from-scratch-with-real-24ff198a1c57>
- [21] Singh, S. (2018). *Understanding The Bias-Variance Tradeoff.*
<https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>
- [22] Amazon Machine Learning. *Model Fit: Underfitting Vs. Overfitting.*
<https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html>
- [23] Listgarten J., Damaraju S., Poulin B., Cook L., Dufour J., Driga A., Mackey J., Wishart D., Greiner R., Zanke B. (2004)
Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms.
 Clin Cancer Res April 15 2004 (10) (8) 2725-2737; DOI: 10.1158/1078-0432.CCR-1115-03
- [24] Ksiazek, W., Abdar, M., Acharya, U. and Plawiak, P., (2019)
A Novel Machine Learning Approach For Early Detection Of Hepatocellular Carcinoma Patients.
<https://www.sciencedirect.com/science/article/pii/S1389041718308714>
- [25] Cancer.Net (2019). *Liver Cancer - Introduction.*
<https://www.cancer.net/cancer-types/liver-cancer/introduction>
- [26] Serag A., Ion-Margineanu A., Qureshi H., McMillan R., Saint M., Diamond J., O'Reilly P., Hamilton P. (2019)
Translational AI and Deep Learning in Diagnostic Pathology.
 Frontiers in Medicine, Volume 6, doi:10.3389/fmed.2019.00185, ISSN:2296-858X
- [27] Brownlee, J. (2016).
Gentle Introduction To The Bias-Variance Trade-Off In Machine Learning.
<https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/>
- [28] National Cancer Institute and World Health Organization. (2009).
HCC in the World. <https://www.bluefaery.org/statistics/>

- [29] Davis J. (2019). *9 Unsecured Medical Databases Found Leaking Sensitive Patient Data.*
<https://healthitsecurity.com/news/9-unsecured-medical-databases-found-leaking-sensitive-patient-data>
- [30] Ray S. (2017). *6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R.*
<https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>
- [31] Zhang Z. (2019). *Naive Bayes Explained.*
<https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0>
- [32] Pant A. (2019). *Introduction to Logistic Regression.*
<https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>
- [33] Deepai.org *What is the curse of dimensionality?*
<https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality>
- [34] wikipedia.org. *Hepatocellular carcinoma.*
https://en.wikipedia.org/wiki/Hepatocellular_carcinoma#Diagnosis
- [35] pl.wikipedia.org. *Rak wtrobowokomorkowy.*
https://pl.wikipedia.org/wiki/Rak_w%C4%85trobowokom%C3%B3rkowy#Badane_potencjalne_markery
- [36] Southern Nevada Health District (2019). *The Five Types of Hepatitis.*
<https://www.southernnevadahealthdistrict.org/Health-Topics/the-five-types-of-hepatitis/>
- [37] Cicalese L. (2020). *Hepatocellular Carcinoma (HCC).*
<https://emedicine.medscape.com/article/197319-overview>
- [38] Geller S. (2019). *Normalization vs Standardization — Quantitative analysis.*
<https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf>
- [39] Sullivan J. (2018). *Data Cleaning with Python and Pandas: Detecting Missing Values.*
<https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b>
- [40] Brownlee J. (2019). *A Tour of Machine Learning Algorithms.*
<https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
- [41] BPU Holdings (2018). *A Tour of Machine Learning Algorithms.*
<https://www.bpuholdings.com/the-importance-of-having-a-good-dataset/>

- [42] Radecic D. (2019). *Feature Selection in Python — Recursive Feature Elimination*.
<https://towardsdatascience.com/feature-selection-in-python-recursive-feature-elimination-19f1c39b8d15>

A: Feature Correlation heat maps, used for feature selection.



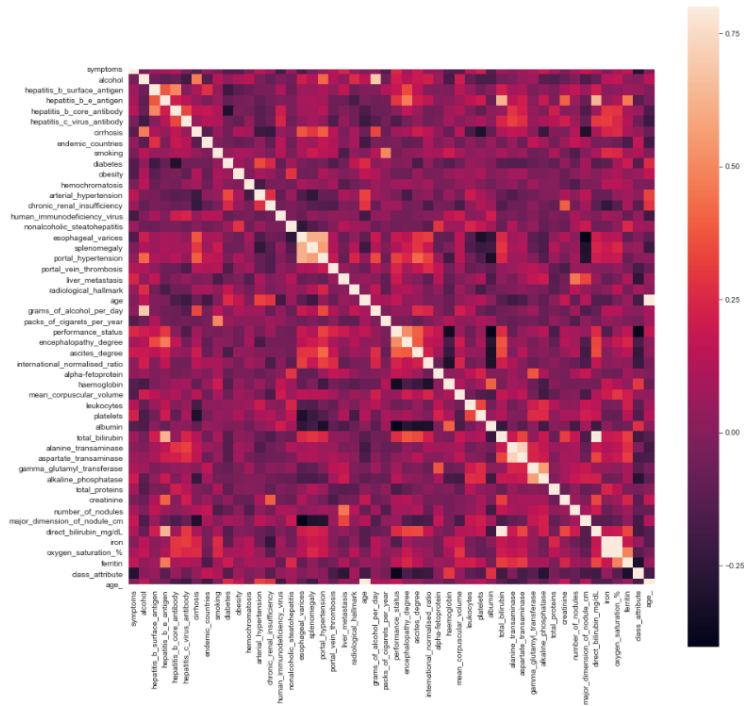


Figure A.2: Correlation of features in the HCC Survival dataset.

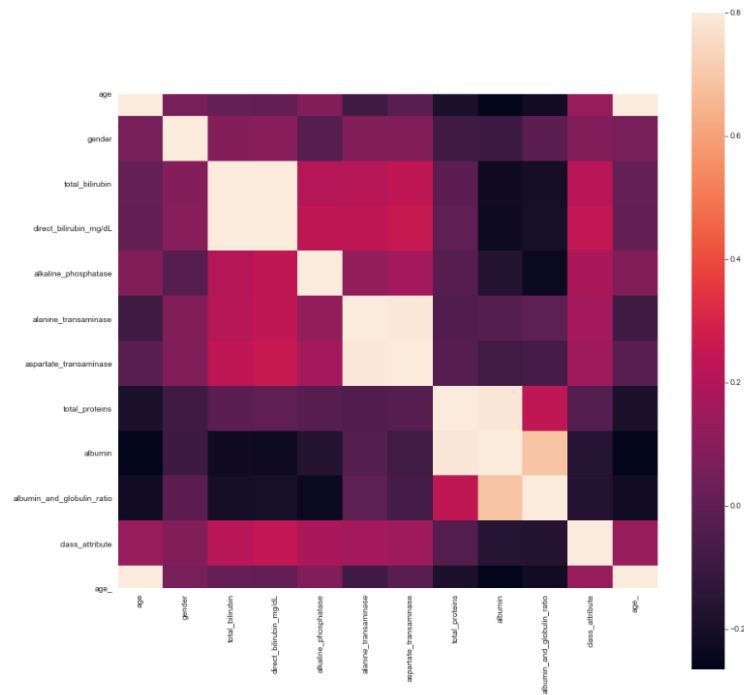


Figure A.3: Correlation of features in the Liver Disease dataset.

B: Predictive Accuracy of Models Built.

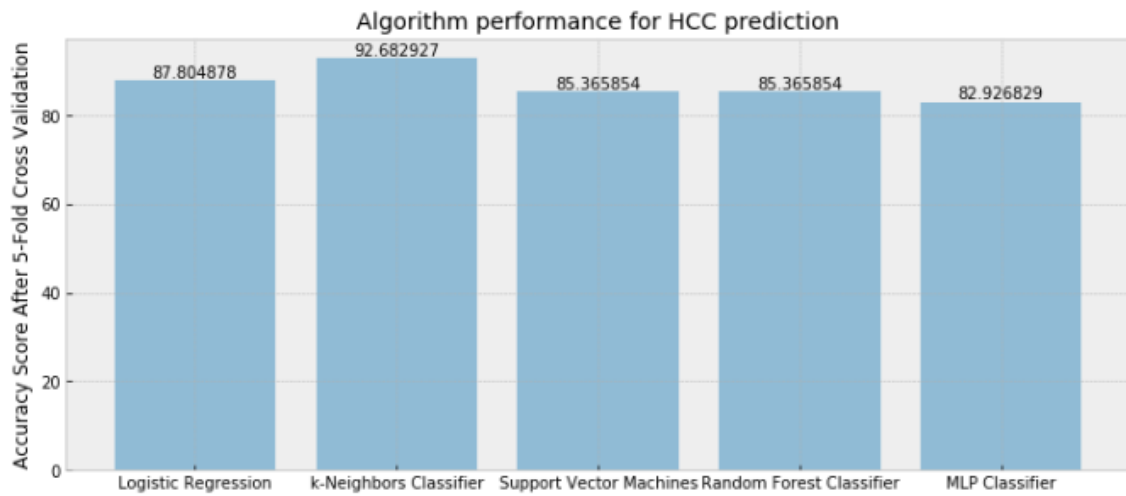


Figure B.1: Accuracy obtained by all algorithms after 5-fold cross validation (HCC Survival).

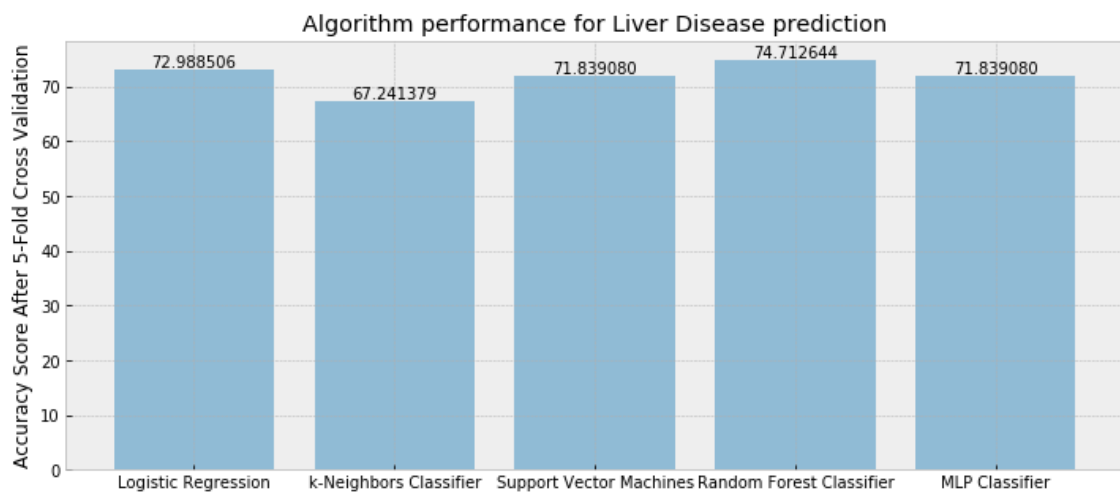


Figure B.2: Accuracy obtained by all algorithms after 5-fold cross validation (Liver Disease).

C: Feature Importance for Liver Disease Dataset.

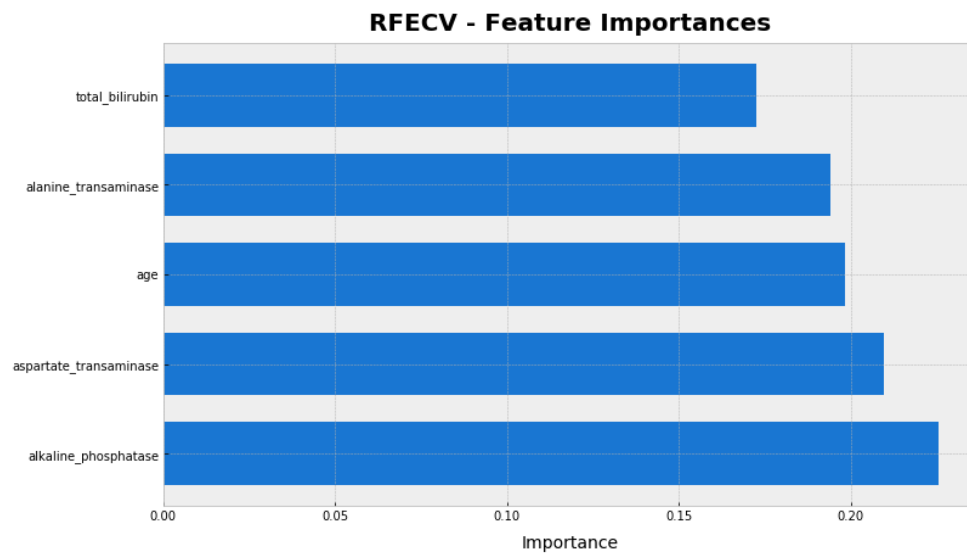


Figure C.1: Ranking the importance of different feature visualized (Random Forest - Liver Disease).

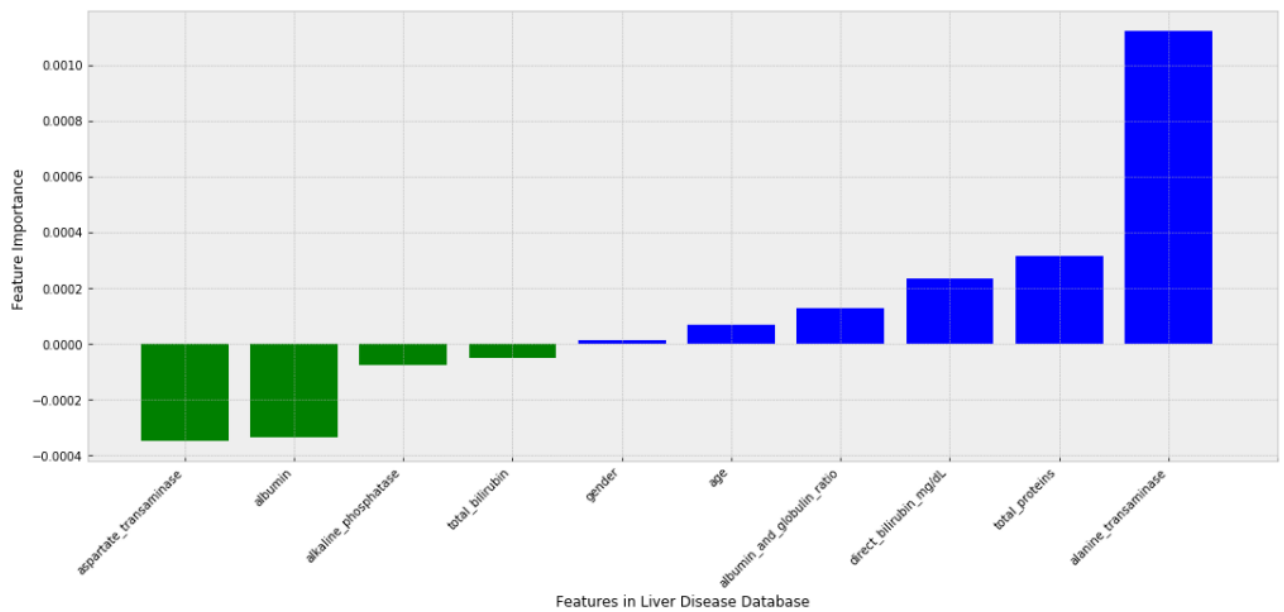


Figure C.2: Ranking the importance of different feature visualized (SVM - Liver Disease).