Feature Selection for Identification of Transcriptome and Clinical Biomarkers for Relapse in Colon Cancer

Lloyd Andreassen

June 1, 2021

Abstract

This study attempts to find good predictive biomarkers for recurrence in colon cancer between two data sources of both mRNA and miRNA expression from frozen tumor samples. In total four datasets, two data sources and two data types, were examined; mRNA TCGA (n=446), miRNA TCGA (n=416), mRNA HDS (n=79), and miRNA HDS (n=128). The intersection of the feature space of both data sources was used in the analysis such that models trained on one data source could be tested on the other. A set of wrapper and filter methods were applied to each dataset separately to perform feature selection, and from each model the k best number of features was selected, where k is taken from a list of set numbers between 2 and 250. A randomized grid search was used to optimize four classifiers over their hyperparameter space where an additional hyperparameter was the feature selection method used. All models were trained with cross validation and tested on the other data source to determine generalization. Most models failed to generalize to the other data source, showing clear signs of overfitting. Furthermore, there was next to no overlap between selected features from one data source to the other, indicating that the underlying feature distribution was different between the two sources, which is shown to be the case in a few examples. The best generalizing models where based on clinical information and second best was on the combined feature space of mRNA and miRNA data.

Contents

1 State of the Art

Machine learning has seen a spike of interest and development in the last decade. Increased computational power combined with easier and cheaper data collection and storage has lead to large datasets being available and a number of different approaches developed to solved a wide variety of tasks. One such field of research is oncology where transcriptome data has become more available and researches are attempting to find good prognostic indicators for survival and cancer recurrence, also known as relapse. This study will focus on the second most prevalent cancer [1], colon cancer, and predicting recurrence. Predicting recurrence is preferred over overall survival given that colon cancer patients are in general quite old, the median age being 73 years in Norway, chapter 2.1 [1]. This means that overall survival can sometimes be problematic given patients die of other causes than the cancer.

The most common prognostic indicators in clinical applications for colon cancer is the Tumor-Node-Metasasis-stage (TNM-stage), which is used to determine follow-up treatment post surgery [24]. Studies have determined other relevant clinical parameters that can be used as prognostic indicators. A recent study found a simple prognostic score based on six clinical parameters for metastasised colon cancer patients [35]. Other studies have implicated DNA Mismatch Repair (MMR) deficiency, or Microsatellite Instability (MSI), correlation with survival in stage 1 and 2 patients [24]. The only molecular markers used in clinical cases as of today are MSI-status [24], BRAF [9], and KRAS [10], where the latter two are only used in metastatic cases.

Prognostic indicators, however, need not be specific clinical variables. Skrede et al. trained a variant of the MobileNetV2 [44] architecture to predict presence of a tumor based on images of stained tumor tissue [49] showing statically significant results for good prognosis post surgery. Guinney et al. [20] considered another approach, performing a Markov clustering of six different predictive models output on messenger RNA (mRNA) data. The study identified four consensus molecular subtypes of colon cancer, each characterised by specific gene expression.

However, this study will focus on gene expression signature, messenger RNA (mRNA) and micro RNA (miRNA), transcriptome data, clinical data, and briefly infiltration estimates [31]. By using the transcriptome counts we can indirectly estimate the presence of proteins that promote cancerous behaviour or suppress anti-cancerous behaviour in cells. A number of different approaches have been attempted, however, each approach has to deal with three core problems. 1) the high dimensionality of the data combined with 2) lower sample size, and 3) class imbalance, about 25% of patients experience. The three problems combined prove particularly challenging and models are prone to overfitting. The following two papers dealt with the high dimensionality by performing single step l1 weight decay regression models on miRNA data to determine a sixteen [23] and four [24] miRNA prognostic indicator, respectively. However, a single weight decay feature selection model is prone to overfitting and selecting biased features. One approach to solve this selection problem is to explore the feature

space with many small trained models, as done in [47]. The researches used a Grasshopper optimization technique to generate small subsets of features that was trained using an SVM model to perform feature selection. This approach is a hybrid method that utilizes a methauristic optimization algorithm.

The process is divided into two steps, first selecting a subset of features based on a number of different feature selectors, then train models on those feature subsets. The feature selection is done with filter and wrapper methods, the latter training models with weight decay regularization to select features. Four classifiers are trained on the subset of selected features on each dataset and tested on the other data source. The goal is to find a set of genes that have high predictive power for determining recurrence, and preferably as small of a set of genes as possible. Smaller subsets of genes lead to less likelihood of overfitting to additional noisy and unimportant features. Furthermore, with fewer genes relevant for the model, it would be easier in a clinical setting to use those genes as a prognostic biomarker, rather than recording gene counts for a large number of possible genes.

Section 2 details the relevant cellular and cancer biology background, in addition to detailing machine learning principles and the models used for the thesis. The methodology, experiments, and datasets will be outlined in section 3. Lastly the results of each of the experiments will be presented in section 4 and possible improvements in section 5.

2 Introduction to Cellular and Cancer Biology and Machine Learning

The following section will detail general principles of Cellular and Cancer Biology and Machine Learning that are relevant background information for the thesis. Section 2.1 will go through the general mechanisms of the Cell Life Cycle, programmed cell death (Apoptosis), and mRNA and miRNA function in protein transcription and translation. The information follows closely chapter 6, 7, and 18 of the following course book in biology [3]. Section 2.2 will briefly mention the hallmarks of cancer, as detailed in [21], and the concept of oncogenes. Section 2.3 will go through the general principles of Machine Learning from what defines a model learning, performance measures, Bias-Varaince tradeoff, and into methods of mitigating said tradeoff in Cross Validation, Regularization, and Feature Engineering. Lastly section 2.4 and 2.5 will introduce the relevant filter methods and classifiers used for this thesis.

2.1 Cellular Biology

Before dwelling into the details of the fundamental mechanisms of cancer, this thesis will detail some basic principles as part of cellular biology that is pertinent to both the discussion of said fundamentals, but also relevant to describe what the datasets for this study. This section will firstly outline the mechanisms of cell proliferation by detailing the different stages, and control mechanisms, of

the cell life cycle and also the mechanism of programmed cell death (apoptosis). Finally a brief explanation of the translation and transcription of proteins and how both messengerRNA (mRNA) and microRNA (miRNA) are part of that process.

2.1.1 Cell Life Cycle

Eukaryotic cells, that is cells with an enclosed nucleus, divides into two daughter cells through a four phase process; Gap 1 (G_1), Synthesis (S), Gap 2 (G_2), and Mitosis (M). However, a cell can enter a resting state (quiescence), known as Gap 0 (G_0), temporarily, or permanently, stopping the cycle. The transition between the four phases are driven by a series of Cyclin Dependant Kinases proteins (Cdk) that are activated when binding to a cyclin protein. Each Cyclin-Cdk pair is associated with promoting a transition from one step to another, for instance the G_1S-Cdk promotes the transition from G_1 to S.

Within a single phase there are checkpoints that need to be passed before the cell continues to the next phase. These checkpoints determine if all preparations has been made for the next phase, and if not halting the transition process. For instance to pass the G_1 checkpoint the cell first checks if there is any DNA damage. If there is some DNA damage, it actives transcription of Cdk inhibiting proteins that halt the transition to the S stage.

In the S phase the cell replicates its DNA and preventing replication from occurring more than once. Due to the structure of the ends of the DNA strands the whole strand cannot be replicated, hence a small section is lost for each replication. Because of this problem, the ends of each chromosome has a set of non encoding genes called Telomeres. The cell has machinery that can detect when its Telomeres is sufficiently degraded and thus stops replicating, acting as a limit on the number of cell divisions a cell can undergo. Section 2.2.2 will go into more detail on the impact of Telomeres in cancer.

2.1.2 Apoptosis

Apoptosis is the process of controlled cell death. Once the apoptosis has started it is irreversible. The process can be initiated either through intrinsic or extrinsic pathways, however, in both cases the same underlying mechanisms are triggered. Enzymes called caspases are activated that dismantle the organelles and proteins in the cell. The cell changes its cell surface, which attracts the attention of specialised phagocytic cells that engulf the remains of the cell. This way the components of the cell does not spill out onto other cells causing an inflammatory response and parts of the cell can be recycled by the phagocytic cell. The intrinsic pathways are generally regulated by the Bcl2 family of proteins while the extrinsic pathways are generally either due to Tumor Necrosis Factors (TNF) or Fas receptors.

Apoptosis is used as a means to control cell proliferation beyond just controlling cell division. Section 2.2.2 will go into more detail on the impact of apoptosis in cancer.

2.1.3 RNA and RNA Transcription

RNA is a single stranded neucleic acid strand molecule that uses the Uracil base instead of Thymine. Given its single stranded structure it can perform more roles than simply storing information because it can fold and form structures. Because of this RNAs can perform a wide variety of different roles like structural, regulatory, and catalytic roles. The RNA strands are transcribed from sections of DNA through the process of DNA polymerases.

2.1.4 Messenger RNA (mRNA)

For the purposes of this thesis the main focus is on protein encoding mRNAs. Those mRNAs are transcribed from DNA as pre-mRNA, then modified and transported to the cytoplasm for translation into proteins. pre-mRNAs consists of introns (non encoding regions) and exons (encoding regions) that is modified prior to translation. The introns are removed and the exons rearranged to form an mRNA molecule that encodes a specific protein. Because the exons can be rearranged, it means that a single gene can encode different proteins depending on how those exons are combined. Each mRNA can be translated into proteins multiple times, depending on the longevity of the mRNA while in the cytoplasm.

2.1.5 Micro RNA (miRNA)

miRNAs are small non-coding RNAs, about 22 nucleotides long, that are packaged with a specialized protein forming a RNA-inducing silencing complex (RISC), which searches for complementary base pair mRNAs in the cytoplasm to induce degradation of the mRNA. The RISC does not, however, get degraded, thus it can search for new mRNAs to silence. This means that miRNA can quite efficiently inhibit translation of certain proteins. However, a single miRNA need not only target a specific mRNA. As long as a mRNA contains the matching sequence a miRNA can block translation from that mRNA.

2.1.6 Estimate of Protein Expression

The mechanisms that govern cell division, apoptosis, and relevant hallmarks of cancer, as discussed in 2.2.2, are controlled by protein expression. Protein expression, however, is more difficult to estimate, hence encoding mRNAs and miRNAs are used as an estimate of the encoded, or silenced, protein. Note that the estimate is not a one to one correlation, given that a single mRNA can encode different proteins depending on how exons are ordered, translate multiple instance of a single protein depending on its longevity, and a single miRNA can silence multiple instances of mRNA molecules.

2.2 Cancer Biology

Cancer is fundamentally a tissue based disease where the regulation of cell proliferation and inhibiting factors leading to continued growth and structural dam-

age to the tissue. This imbalance in regulation comes from several different biological mechanisms that take place in a cell or part of intercellular processes. This section will detail some of the important mechanisms that explains how cancer develops, grows, and spreads, and introduce the specific cancer type relevant for this thesis, namely Colon Cancer.

2.2.1 Oncogenes

Genes that promote cancerous behaviour after being mutated are referred to as oncognees. An oncogene could for instance promote cell growth or inhibiting apoptosis. Most oncogenes begin as a proto-oncogene, a gene that could, once activate, act as an oncogene. It is through the protein encoding of these genes that the cancerous behaviour is expressed. For instance p53 regulates DNA damage repair, however, a mutation in the encoding for p53 has been shown to be significant in different types of cancers.

Proto-oncogenes can become oncogenes with only a small modification in its original function. There are three primary ways of activation; mutation, increased expression, and chromosomal translocation. A mutation in the proto-oncogene could change the structure of the encoded protein causing a loss in its original regulation. Increased expression could come from interactions with other proteins or for instance downregulation of certain miRNAs that down-regulate said protein. Lastly chromosomal translocation involves the specific gene being translocated to a different region and/or merged with another gene, which could lead to a change in expression. In addition to oncogenes there are genes coined anti-oncogenes, or tumor suppressor genes, that encode regulation of cell division and survival that promote cancerous behaviour by being downregulated.

2.2.2 Hallmarks of Cancer

The following paper from 2011 details the current known 6 biological hallmarks that define cancer in addition to outlining two possible emerging hallmarks and two enabling mechanisms [21]. The following section will briefly introduce these key hallmarks of cancer.

A tumor requires obtaining sufficient signals to promote, and continue, cell division, spreading, and to evade apoptosis. The hallmarks of cancer revolve around these two fundamental principles. Firstly, as detailed in section 2.1.1, cells cannot divide forever as their telomere genetic code gets shortened for every DNA sythesis the cell goes through. Additionally, the protein telomerases can prolong the period for which a cell can continue to divide without shrinking its telomeres. This replicative immortality is a core hallmark of cancer and considered a necessary condition for cancer development. Without this feature, a cell would either go into senescene, and stop replicating, or undergo apoptosis. It has been shown that shortening the telomeres in mice have a direct correlation to reducing the risk of cancer development and that the lack of telomerases may prevent neoplastic development past a microscopic state. However, the

telomerases protein does not only function as a means of providing replicative immortality, as it has been shown to have an impact on enhancing cell proliferation, resisting apoptosis, DNA damage repair system, and RNA polymerases function for transcription of RNA.

However, the immortalized cells still need to stimulate proliferation and evade growth suppressing functions. Both can occur via an intrinsic pathway or get extrinsic pathway from other cells, for instance the supportive stromal cells. Growth suppressor factors are part of the cell life cycle control system outlined in section 2.1.1, namely the cyclin-Cdks, where the upregulation of specific cycline-Cdks have a direct impact on the progression through the cell phases for cell division. For instance, the protein p53 suppress the continuation to the S phase when there is a sufficient amount of DNA damage in the cell, and in extreme cases inducing apoptosis provided the damage is not repairable. p53 can also react to other stress and abnormalities in the cell function. On the other hand retinoblastoma (RB) is a external growth suppressors that also impacts cell proliferation, activating senescene, or inducing apoptosis. Studies have shown that the growth suppressing functions have a degree of redundancy, as Rb negative and p53 negative mice, that is mice without the presence of the gene encoding the relevant protein, developed normally, however, experiencing abnormal developments later in life.

To sustain this continued growth, the tumor needs nutrients and energy. To facilitate the continued expansion, the process of angiogenesis, development of new blood vessels, have been shown to be of vital importance. An inducer of angiogenesis is vascular endothelial growth factor-A (VEGF-A) that encode the development of new blood vessels and the homeostatic survival of endothelial cells. However, the produced vessels by angiogenesis are abnormal, usually containing convoluted and excessive branching, distorted and enlarged vessels, erratic flow, and leakiness. Studies on mice have shown that upregulation of angiogenesis inhibitors impair tumor growth, while a downregulation increases growth of both planted and naturally developed tumors.

In addition to angiogenesis to sustain continued growth, cancer cells reprogram their metabolism to support further growth. Ordinarily glycolysis is used in anaerobic metabolism, however, cancer cells use glycolysis despite working under aerobic conditions, leading to a state of "aerobic glycolysis". This change in metabolism leads to a drastic reduction in efficiency of ATP production, that the cell offsets partly by upregulation of glucose transporters, however, the glycolysis servers other purposes as well. It is hypothesised that the glycolytic is diverted to biosythetic processes that generate nucleotides and amino acids, facilitating more DNA synthesis.

Beyond growing, cancer tumors need to resists induced cell death. Section 2.1.2 introduce the general principles of apoptosis, which is induced either through intrinsic (Bcl-2 family of proteins) or extrinsic pathways (Fas ligand/recepters or TNF). Protein p53 is a tumor suppressor gene that is associated with DNA damage that halts transition from G_1 to S stage of cell divison, but it can also induce apoptosis if there is too much DNA damage to repair. Limiting the presence of p53 is the most common strategy to resist induced apoptosis. It has

also been shown that the intrinsic pathway is more widely implicated in halting carcinogenic development over the extrinsic pathway.

Lastly tumors can spread to other tissue either through invasion of nearby tissue or metastasis to spread to distant tissue through the circulatory or lymphic system. A key component for invasion or metastasis is "colonization", the process in which a microscopic tumor can grow within the new environment to a macroscopic tumor. Provided there is no facilitating growth factors, or other microenvironments, that cancer cells require to pass beyond microscopic tumor, they may revert to a noninvasive state. That is the metastasis have been able to "physically disseminate", however, unable to "adapt" to the foreign environment. Alternatively the microscopic tumor could be dormant and erupt later after the primary tumor has been dealt with. However, in the early stages of the invasion-metastasis cascade, a multistep process detailing invasion and metastasis progress, proteins or supporting cells can have an impact on the process of infiltrating new tissue. For instance E-cadherin, a cell to cell adhesion molecule, that creates a cell sheet of quiescence cells limiting invasion and metastasis. It has been show that downregulation and mutational inactivation of E-cadherin is present in human cancers. On the opposite end stromal cells, through secreting CCL5, can stimulate invasive behavior. Similarly a buildup of inflammatory cells near its boundary of a tumor can produce the necessary enzymes for invasion, such that the cancer cells need not produce the activating proteins of the ephithelial-mesenchymal transition (EMT) program, a regulatory program that is casually important for invasion and metastasis and resisting apoptosis.

Of the key hallmarks mentioned above, the paper [21] notes two enabling hallmarks that facilitate the expression of those mentioned. The first being a change in metabolism to support further growth. Ordinarily glycolysis is used in anaerobic metabolism, however, cancer cells use glycolysis despite working under aerobic conditions, leading to a state of "aerobic glycolysis". This change in metabolism leads to a drastic reduction in efficiency of ATP production, that the cell offsets partly by upregulation of glucose transporters, however, the glycolysis servers other purposes as well. It is hypothesised that the glycolytic is diverted to biosythetic processes that generate nucleotides and amino acids, facilitating more DNA synthesis. Secondly is genome instability and mutations that directly impact the underlying regulatory mechanisms/pathways and proteins for each of the hallmarks. During the process of tumorigenesis cancer cells often increase the rates of mutations, one of the most common ways by downregulating p53. Additionally, the mutations can compromise the cell control system that leads to apoptosis or senescence.

2.2.3 Colon Cancer

Colon cancer (CC), or colorectal cancer, is a cancer developed in either the colon or rectum that ranks as the second most prevalent cancer among women and third among men, the median age being 73 in Norway, chapter 2.1 [1]. There are many risk factors, among which are age, dietary, and obesity, with a about 15-30% being hereditary having some major hereditary component [17]. Despite

comprising of a small portion of the total patient population, the heraditary cases are studied to understand the underlying mechanism better. Of note is the mutation in the Adenomatour Polyposis Coli (APC) tumor suppressor gene is a contributing factor in Familial Adenomatous Polyposis (FAP), a subtype of colon cancer that develops at a young age [17]. Others gene mutations like KRAS [10], BRAF [9], PIK3CA [38], and TP53 [17], the gene that governs the p53 protein, have also been shown to be relevant biomakers for colon cancer. Together APC, KRAS, BRAF, and TP53 account for nearly 70% of all colon cancer cases [2].

A key prognostic factor of colon cancer is the Tumor, Node, and Metastasis Staging (TNM-Stage). The staging consists of four main stages dependant on the independent Tumor, Node, and Metastasis stage. Principally TNM stage 1 is defined by small and local tumors. Stage 2 defined by not having spread to lymph nodes. Stage 3 is defined by spread to lymph nodes. And stage 4 is defined by metastasis.

The tumor stage differentiates between how many specific layers of tissue has been penetrated by the tumor, ranging from growing through the inner lining, muscle, outer lining, and further into a different organ. The node stage differentiate how many lymph nodes the cancer has spread to. Lastly the metastasis stage differentiate if the tumor has spread to a distant organ or not. As mentioned above, any metastasis means that the TNM-stage is stage 4. A more detailed explanation of the TNM-staging can be found in [13].

Standard treatment for colon cancer is surgery to remove the primary tumor. Most patients are cured by the surgery, however, some develop recurrence. Adjuvant chemotheraphy is a possible followup treatment aimed at eradicating micrometastases, chapter 9.6 [1]. In Norway stage 3 patients are treated with adjuvant chemotherapy, however, only specific high risk stage 2 patients get the same treatment. The type of chemotheraphy is a combination of fluorouracil (5-FU), folinat (FLV-regimen) and oxaliplatin (FLOX), however, 5-FU based chemotherapy is not used for MSI-H patients. Research has shown that MSI-H patients have no effect on those patients [45, 46].

2.3 Machine Learning

The topic of Machine Learning is dedicated to problem solving by replicating/simulating the process of "learning" based on observations, data, and/or predetermined knowledge. Learning, in this context, is the process of remembering information, adapting the information to solve a problem, and generalizing that knowledge to an unseen circumstance. Humans are capable of doing this process all the time. Find a picture and description of important characteristics of an animal, and a person might be able to recognize the animal if they encounter it later. Alternatively, people learn how to distinguish people they know from others they do not. For a computer, on the other hand, finding a way to replicate this process can be quite difficult given that one has to define how to modify and adapt to new knowledge. Many different methods have been proposed to simulate this process of learning based on data, ranging from

something as simple as finding the line of best fit, linear regression, for a given number of sample points, to complex convolution neural networks performing image segmentation. Some models are iterative, learning by repetition on the data, while others not. Additionally, a number of different training schemes are used in conjunction with different methods, leading to a large possibility of available options.

This following subsection will detail a number of key concept related to machine learning. Firstly the basics of handling datasets and then the concept of different types of machine learning. Next the key concept of Overfitting/Underfitting and its relation to the Bias-Variance Tradeoff will be explained. Given the importance of reducing bias and variance, the two following sections will detail two possible ways of achieving that, namely Regularization and Cross-Validation, respectively. Section 2.3.6 outlines a method to search over a large hyperparameter space for methods that have hyperparameters that impact the models performance. Lastly, quite pertinent to the dataset that is used in this thesis, the concept of Curse of Dimensionality and Feature Engineering is introduced in section 2.3.7 and 2.3.8, respectively.

2.3.1 Process of Learning

The process of learning from data mentioned above consists of a number of key elements, and may be summarised as follows; "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E", chapter 5 [18]. The experience E, is a set of data samples x_i consisting of a k features $x_i = (x_{i,1}, ..., x_{i,j})$. The exact shape of the feature space need not be, as outlined above, a single vector. For instance. an image has a three dimensional feature space consisting of a pixel width, pixel height, and color depth. Alternatively, a single sample could be the string of a sentence instead of a dedicated feature space, however, in language based models, a preprocessing step includes converting a sentence to a set of features. For the purpose of this thesis the feature space will be considered as a one dimensional space of k features. In cases where a feature is categorical, the feature will be converted to a set of binary features through a one hot encoding, thus each feature will be treated as a real number.

The data that is used is generally split into three components, the training, test, and validation sets, respectively. Each dataset serves a specific role in the learning process. Firstly, the machine has to learn from something, and this is the training set. To know that a model has learned something, and can apply it to unknown samples, we use a test set, checking the strength of the generalization of the model. However, a simple split between training and testing need be sufficient for all use cases, hence the introduction of the validation set. Consider a model like K-Nearest-Neighbours that has a parameter k, the number of closest points to a point x_i is used to determine the classification of x_i. The model output can drastically change depending on what value of k is selected, hence a validation set will be used to determine the most optimal selection of k. It is

important to note that the validation set is separate from the test set, as the validation set's primary role is to determine the relative performance of different choices for a model or iterative steps as a model trains. For instance for a neural network, after each epoch, it validates the current trained model on a validation set to determine how well it has generalized so far. A common approach for validating during training, or for selecting different parameter choices, will be detailed in section 2.3.5.

The task T can vary wildly. Broadly speaking it can be categorised into Supervised, Unsupervised, and Semi Supervised/Reinforcement learning. For supervised learning each data sample has an associated target value y_i such that each element in X is of the form (x_i, y_i). The target/response/class determine the associated class or value of a specific sample, and the learning process aims to learn to predict the target of new data samples. In the case of a categorical target the problem is classification, while for continuous variables it would be a regression problem. Other types of problems, like image segmentation have a different response, however, for the purpose of this thesis, classification and regression is the important sub-tasks for supervised learning. Unsupervised learning, on the other hand, has no target, hence the models aim to learn some underlying structure in the data. For instance, clustering of data samples that are similar or learning some lower dimensional representation of the data. Methods of dimension reduction will be mentioned in more details in section 2.3.8. Semi-supervised learning is a hybrid approach between supervised and unsupervised learning. In cases where recording data samples is cheap, but recording targets is expensive, models can be trained on a smaller set of labeled data in a supervised way, then trained on a larger unlabeled dataset in an unsupervised way. Alternatively a model could consist of multiple smaller models that act in conjunction to learn on unlabeled data in a teacher-tutor relationship or only know that its prediction is wrong, but not what the correct prediction is, usually referred to as learning by critic. Given the nature of the problem for this thesis, only supervised learning will be discussed, with some minor mentions of unsupervised learning as a means to visualize data in lower dimensions.

The performance measure, P, will be explored in more detail in the following section.

2.3.2 Performance Measure P

A measure P is necessary to determine if a model has learned. The measure used depend on the type of task being evaluated, but is generally a dissimilarity measure.

For regression models the Mean Square Error (MSE) is normally used, chapter 2.2 [25], which is simply $\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$ for some model function $\hat{f}$ estimating an underlying true distribution f. However, the choice of MSE is not arbitrary. To see this consider the principle of maximum likelihood. For an arbitrary probability distribution $P(X|\theta)$ the maximum likelihood is the choice of θ such that $\theta_{ML} = \arg\max_\theta P(X|\theta)$. For certain probability distributions $P(X|\theta)$ it can be more useful to consider the log likelihood, i.e. instead maximize

$\theta_{ML} =_\theta \log\big(P(X|\theta)\big)$.

Now consider the case of a linear regression model. Let $\hat{f}(x;\theta)$ be the regression model where θ is the set of learned polynomial coefficients including a bias term. Given that the goal of the model is to predict a response y based on f, instead the correct formulation is the maximum likelihood of the conditional probability distribution $\arg\max_\theta \ell(\theta) = \arg\max_\theta P(Y|X;\theta,b)$. Assume that the noise of the data is Gaussian with mean zero, i.e. each sample response y is of the form $y = f(x) + \epsilon$ for $\epsilon\ \mathcal{N}(0,\sigma^2)$. Given a linear case the log likelihood of the conditional distribution can be expressed as follows;

$$\begin{aligned}
\ell\ell(X,\theta) =& \sum_{i=1}^{n} \log\mathcal{N}(y_i - \boldsymbol{\theta}^T\mathbf{x_i}\,|\,\mathbf{0},\sigma^{\mathbf{2}}),\\
=& -m\log\sigma - \frac{m}{2}\log(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\theta}^T\mathbf{x_i})^{\mathbf{2}},\\
=& -\text{const} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\theta}^T\mathbf{x_i})^{\mathbf{2}}, \qquad (1)
\end{aligned}$$

where the last term is equivalent to the sum of squared difference between the target and the predicted target, see chapter 5.5 [18]. The maximum likelihood solution to a regression problem could be solved analytically from (1), however, more complex models would be more difficult, or impossible, to solve analytically. Additionally, it is entirely possible that the noise in the data is not Gaussian, hence the log likelihood would be different.

The maximum likelihood approach comes naturally if we consider the Kullback-Leibler divergence;

$$\begin{aligned}
D_{KL}(p\ ||\ q) =& H_p(q) - H(p),\\
=& \mathbb{E}_{x\ p}[\log p(x) - \log q(x)], \qquad (2)
\end{aligned}$$

of two distributions q and p, with x sampled from q, chapter 5.5 [18]. $H_p(q)$ is the cross entropy between p and q and $H(p)$ the entropy of p. If p is the underlying data generating distribution and q is the model distribution, then minimization of the KL-divergence is the same as maximizing the log likelihood given that;

$$\begin{aligned}
\arg\min_\theta D_{KL}(p_{data}\ ||\ p_{model}) &= \arg\min_\theta \mathbb{E}_{x\ p_{data}}\big[\log p_{data}(x) - \log p_{model}(x)\big],\\
&= \arg\min_\theta \mathbb{E}_{x\ p_{data}}[\log p_{model}(x)],\\
&= \arg\min_\theta \log\prod_{i=1}^{n} p_{model}(x_i),\\
&= \arg\min_\theta \ell\ell(X;\theta).
\end{aligned}$$

Thus using maximum likelihood estimate for a model distribution is an attempt at making the model distribution and the data distribution as similar

as possible. This approach would be ideal if we could have access to the data generating distribution, however, that is the distribution p_{model} is attempting to estimate. Instead we will only be able to make the model distribution as close to the distribution of the available data as possible, however, it still remains a good approximation. Many machine learning methods will use KL-divergence indirectly due to other cost functions, or directly.

One approach to classification is to perform regression and base the predicted response based on $\hat{f}(x) = \mathbf{w^T x}$. A simple rule could be to classify all samples above 0 as class 1 and all samples below 0 as class 0 in a binary classification problem, however, this hard classification boundary might not be ideal given the uncertainty near the boundary. This is the principle approach for the Support Vector Machine algorithm which will be explained in section 2.5.1. Alternatively the regression response could be transformed via a function such that the decision boundary becomes smooth. A standard function for this kind of problem is the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$. With this function the probability of a sample being classified as class 1 is $P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w^T x})$, chapter 4.3 [25]. It can be shown that the maximum likelihood estimate for the weights $\mathbf{w}$ is the binary cross entropy, that is;

$$\begin{aligned}\ell\ell(X, \theta) &= \log \prod_{i=1}^{n} P(y_i = 1 \mid x_i)^{y_i} (1 - P(y_i = 1 \mid x_i))^{1-y_i}, \\ &= \sum_{i=1}^{n} y_i \log \sigma(\mathbf{w^T x_i}) + (\mathbf{1} - \mathbf{y_i}) \log \big(\mathbf{1} - \sigma(\mathbf{w^T x_i})\big), \end{aligned} \tag{3}$$

which is the binary cross entropy of a Bernoulli distribution.

However, the performance measure P need not be directly linked to the underlying cost function that defines the optimal learning procedure. For instance accuracy is a simple measure of how many samples are predicted correctly. Simply it is the expectation of the function $I(y_i, \hat{y}_i)$ that has a value 1 when $y_i = \hat{y}_i$ and zero otherwise. Or in simpler terms, the number of correctly predicted samples divided by the total number of samples. Models that have a higher prediction accuracy perform better at classification in theory, however, the accuracy measure might be misleading. To see why, consider a dataset with a class imbalance where 90% of the samples are of class 1 while the remaining 10% are of class 0. In this case a model that always predicts all samples to be 1 would get a 90% accuracy score, yet it has clearly not learned anything meaningful in separating the two classes. An alternative approach to measuring accuracy is the balanced accuracy [7] measure defined as;

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right), \tag{4}$$

for a binary classification problem where TP, FN, TN, and FP are related to the concept of a confusion matrix, seen in table 1. This performance measure penalizes models that get good accuracy based on the class imbalance, giving

	Prediction: False	Prediction: True
Label: False	True Negative	False Positive
Label: True	False Negative	True Positive

Table 1: The columns are the predicted class and the rows are the true labels y. Each cell records the number of samples with the ordered pair of labels $(y, \hat{y})$. Each cell in this example of two classes have a name. True Negative (TN) is the number of samples that are correctly predicted as false. True Positive (TP) is the number of samples that is correctly predicted as true. Both False Negatives (FN) and False Positives (FP) are the cases where the prediction is different from the true label.

them a balanced accuracy of 50% in a binary classification problem, or specifically $\frac{1}{c}$ where c is the number of classes. The strength of a models performance is how far it deviates from the baseline of 50%, which will be discussed in section 4.

However, in the class imbalance case above, we could instead measure how much it predicts class 0 as 1 and class 1 as 0. This can be shown in a confusion matrix [16]. A confusion matrix is an $C \times C$ matrix for number of classes C, where each row is the true class and each column is the predicted class. Each cell is the number of occurrences of the pair $(y_i = c_a, \hat{y}_i = c_b)$ for some classes c_a and c_b. The accuracy measure mentioned above is simply the diagonal sum divided by the total number of samples when viewed as a confusion matrix.

Consider a two class problem confusion matrix shown in table 1. One way to include the rate of incorrect predictions in a scoring metric is using a Receiver Operating Characteristic (ROC) curve [16]. An ROC curve plots the False Positive Rate (FPR) against the True Positive Rate (TPR), calculated by $FPR = \frac{FP}{TN+FP}$ and $TPR = \frac{TP}{FN+TP}$. Note that inherent in the computation of the TPR and FPR, the algorithm adjusts the threshold of the probability of a true sample. For instance, in the logistic regression example a probability of $P(y_i = 1 \mid x) > 0.5$ is classified as 1, however, this does not have to be the optimal choice of a threshold for a given model. In this way, the ROC curve finds the most optimal threshold, and from that threshold, computes the TPR and FPR. The final curve should be above the diagonal curve from $(0, 0)$ to $(1, 1)$, also called the line of chance, with a better model being closer to the top left corner. Models that fall below the line of chance need not be worthless, considering such models are simply predicting true samples as false and false samples as true. Switching the prediction output will yield a model that is above the line of chance. A measure of a ROC curve is the Area Under the Curve, which is simply the integral of the ROC curve [16]. Thus the worst possible AUC is 0.5, given that all curves with AUC less than 0.5 can be switched to above 0.5 with the trick mentioned above.

For the purpose of this thesis, the imbalanced accuracy measure and AUC will be used as performance measures.

2.3.3 Bias-Variance Tradeoff

Consider a simple regression problem where the variable y is dependent on x via an unknown functional relationship $y = f(x) + \epsilon$ where ϵ some noise with mean 0 and variance σ^2. Without loss of generality the mean of the noise can be assumed to be 0, for if it was not, it could be considered as a constant term in the function f. As described in section 2.3.2 the normal loss function for regression is MSE. Consider the expected error of any given point x outside the training set for some function $\hat{f}$ trained on D, that is $\mathbb{E}_D[(y - \hat{f}(x; D))^2]$. This decomposes into three terms called the bias, variance, and irreducible error, three key concepts when analysing a models performance. The derivation of the exact relationship is detailed below.

For the purpose of notation, the expectation outlined below will be assumed to be over the domain D of the training set unless specified otherwise. the functions $f(x; D)$ and $\hat{f}(x; D)$ will be abbreviated as f and $\hat{f}$ to simplify the notation. Thus by the definition of expectation we can write;

$$\mathbb{E}[(y - \hat{f})^2] = \mathbb{E}[(f + \epsilon - \hat{f} + \mathbb{E}[\hat{f}] - \mathbb{E}[\hat{f}])^2].$$

Expand and apply linearity

$$\begin{aligned} =&\mathbb{E}[(f - \mathbb{E}[\hat{f}])^2] + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[\epsilon(f - \mathbb{E}[\hat{f}])] + \mathbb{E}[(\mathbb{E}[\hat{f}] - \hat{f})^2] \\ &+ 2\mathbb{E}[\epsilon(\mathbb{E}[\hat{f}] - \hat{f})] + 2\mathbb{E}[(f - \mathbb{E}[\hat{f}])(\mathbb{E}[\hat{f}] - \hat{f})], \end{aligned}$$

Given that f and $\mathbb{E}[\hat{f}]$ are deterministic, that is they are independent of the domain of expectation D, the expectation of a product is the product of expectations. Additionally, since ϵ is independent of all other variables, its expectation can be multiplied out by the product rule, that is if X and Y are independent random variables, then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$. Hence the above expression simplifies to;

$$\begin{aligned} =&(f - \mathbb{E}[\hat{f}])^2 + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[\epsilon]\mathbb{E}[(f - \mathbb{E}[\hat{f}])] + (\mathbb{E}[\hat{f}] - \hat{f})^2 \\ &+ 2\mathbb{E}[\epsilon]\mathbb{E}[(\mathbb{E}[\hat{f}] - \hat{f})] + 2(f - \mathbb{E}[\hat{f}])\mathbb{E}[(\mathbb{E}[\hat{f}] - \hat{f})], \end{aligned}$$

furthermore, given that ϵ has a mean of zero, its expectation is by definition zero, cancelling out the epsilon expectation terms. The last term cancels given that by linearity, $\mathbb{E}[\mathbb{E}[\hat{f}] - \hat{f}] = \mathbb{E}[\mathbb{E}[\hat{f}]] - \mathbb{E}[\hat{f}]$, which is simply zero given that the expectation of an expectation is the expectation, i.e. $\mathbb{E}[\mathbb{E}[\hat{f}]] = \mathbb{E}[\hat{f}]$. Thus,

$$\begin{aligned} =&(f - \mathbb{E}[\hat{f}])^2 + \mathbb{E}[\epsilon^2] + (\mathbb{E}[\hat{f}] - \hat{f})^2, \\ =&\text{Bias}[\hat{f}] + \text{Var}[\epsilon] + \text{Var}[\hat{f}], \end{aligned} \tag{5}$$

which is the outcome outlined above. The simplification from the last line comes from the definition of variance, which by simple expansion and application of linearity shows that, $\text{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$ for some random variable X.

Similarly for the ϵ term, since $\mathbb{E}[\epsilon] = 0$, then $\text{Var}[\epsilon] = \mathbb{E}[\epsilon^2] - (\mathbb{E}[\epsilon])^2 = \mathbb{E}[\epsilon^2]$. This relationship between the variance and bias of the model to the error is key to understanding how to create a model of best fit for a given dataset. The variance of the error, however, is the underlying error that comes from some unknown influence on the variable y. For instance, it could be due to a number of missing variables that would adequately explain the relationship, or simply that the underlying mechanism has some inherent unpredictability. Regardless of its source, it serves as a lower boundary for the mean square error on any dataset, thus the loss of the cost function will never get to zero.

The variance represents the error related to the subsampling of data that is done for the training set D compared to the true distribution. Thus smaller datasets tend to suffer heavily from variance and increases the dataset size reduces the impact of variance. The bias is the error from the inherent assumption built into the model for the function $\hat{f}$. More complicated models tend to reduce bias while simple models increase the bias. For instance a model like linear regression has a low complexity, thus high bias, from the limited number of distributions it can fit. On the other hand polynomial regression will have a higher degree of complexity and infer less bias.

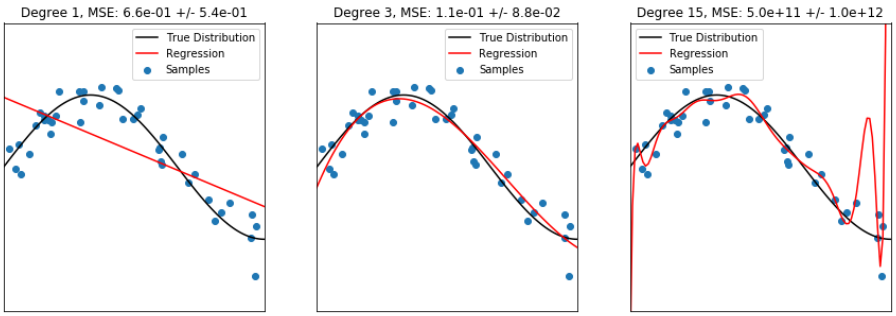

Figure 1: Shows the difference in line of best fit for polynomial regression on the function $f(x) = \sin(1.5\pi x) + \mathcal{N}(0, 0.2)$ for $x \in [0, 1]$

This example can be seen in figure 1. The figure shows the true distribution $f(x) = \sin(1.5\pi x)$, the 42 samples from said distribution with an added noise term of $\mathcal{N}(0, 0.2)$, and the polynomial regression. The figure on the left shows a first degree regression model, right shows a 15th degree regression model, and the middle shows the degree that minimizes the mean square cross validation error between 1 and 15. The linear regression model cannot hope to fully capture the variability in the true distribution given its low model complexity, leading to a high bias term. The 15th degree regression model suffers from high variance and is clearly a poor fit given the right tail end of the regression line. The clearly poor fit of both the 1st and 15th degree polynomial regressions are examples of underfitting and overfitting, respectively, chapter 5.2 [18]. The connection between overfitting/underfitting and model complexity is shown in figure 2.

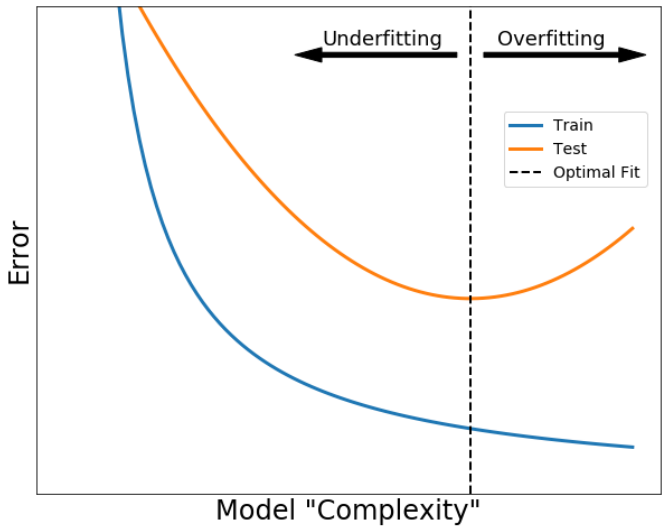

Figure 2: Shows a sample training and test error plotted against model complexity. Models that are the the left of the optimal minimum test error is considered underfitted, while models to the right of that line is considered overfitted.

2.3.4 Regularization

Models can have a high complexity leading to fitting to the underlying noise of the data, as described in the section above 2.3.3. One way to remedy this overfitting is to use a regularization method that constrains the possible model complexity. Consider a polynomial regression model with degree k using mean square error as the cost function. Let each coefficient be β_j. This regression model will then be used to estimate some polynomial function, f, of degree less than k. In this case the model that is used has too high model complexity and will overfit to the underlying noise, especially so with few samples. However, if the model could just zero out specific coefficients, it could express polynomials of lower degrees without the possibility of having too high model complexity. One approach could be to force the coefficients β_i as close to zero, thus if a particular power is not necessary to estimate the underlying polynomial function, the coefficient will be set to zero. This is the l2-norm, or in terms of polynomial regression, a ridge regression model, chapter 6.2 [25]. Mathematically, the problem can be expressed in terms of an optimization problem as follows;

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i) \right)^2 \right\} \quad \text{subject to} \quad \sum_{i=1}^{n} \beta_j^2 \leq s, \tag{6}$$

which amounts to an addition of a $\lambda \sum_{j=1}^{n} \beta_j^2$ term to the cost function. This means that the model is penalized for having high coefficient magnitudes, and

thus irrelevant coefficients will tend to zero. l2-norm forces coefficients close to zero, however, not exactly to zero. In certain situations, like when doing Feature Engineering, as discussed in section 2.3.8, forcing the algorithm to set coefficients equal to zero can be beneficial. One way to achieve this is using the l1-norm instead, that is add a $\lambda \sum_{j=1}^{n} \left|\beta_j\right|$ term to the cost function. This is equivalent to the following optimization problem;

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i) \right)^2 \right\} \quad \text{subject to} \quad \sum_{i=1}^{n} \left|\beta_j\right| \leq s. \tag{7}$$

In terms of regression, this is called a LASSO regression model, chapter 6.2 [25]. The reason the coefficients are forced to zero, compared to a Ridge model, comes from how the restriction of the coefficient space intersects with the level sets of the cost function. An example of this is shown in figure 3 where the red concentric circles are level sets of a cost function, green circle is the l2-norm restriction, blue diamond is the l1-norm restriction, and the axis are two coefficients β_1 and β_2 for some model with two coefficients. The intersection between the level set of a cost function and the restriction function on the coefficient space is the solution model. Given this, the shape of a l2-norm has a higher chance to intersect with the level set away from the axis compared to an l1-norm. However, one could use a hybrid approach between these two restrictions, i.e. a linear combination of the two, which leads to what is called elastic-net regularization, by adding the a $\lambda_1 \sum_{j=1}^{n} \beta_j^2 + \lambda_2 \sum_{j=1}^{n} \left|\beta_j\right|$ for two mixing coefficient λ_1, λ_2. The visual representation of elastic-net regularization in two dimensions is a shape between the green circle and blue diamond in figure 3.

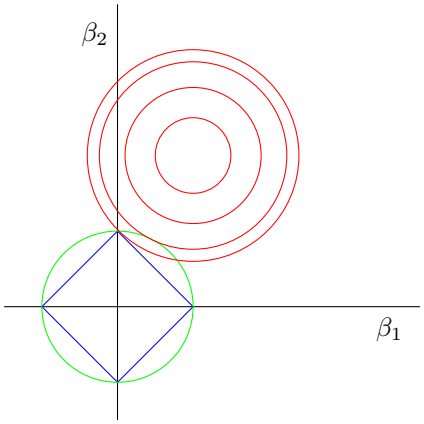

Figure 3: The figure shows a simplified picture of the intersection of the level sets of a cost function, red concentric circles, with l1 (blue) and l2-norm (green) restriction on the coefficient space in two dimensions. Each axis represents one coefficient, i.e. β_1 and β_2. Each red circle represents a boundary of equal loss, i.e. a level set of the cost function, and circles are used as a simplification, given that the level set can be a much more complex boundary. The intersection between a red circle and either the green or blue shape represents the solution a model finds. Take the l2-norm, the second most outer level set intersects with the green circle, forcing the coefficients to be small, yet not zero. For l1-norm, the intersection instead occurs at an axis intercept. In general the intersection for l1 would occur at the axis intercept, which forces coefficients to zero.

The three types of regularization mentioned above are examples of weight decay regularization and can be used for more complicated models than polynomial regression. Support Vector Machine, as is discussed in section 2.5.1, is one prominent example that is relevant to this thesis, however, it could also be applied to individual layers of a neural or convolutions neural network, chapter 7.1 [18]. However, regularization is a more general principle than simply weight decay. Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error, chapter 5.2 of [18]. With this in mind a number of different methods can be considered regularization beyond just weight decay. For instance, early stopping of iterative algorithms, i.e. neural networks and stochastic gradient descent classifier, parameter sharing, i.e. convolutions network architecture, or bagging/ensemble methods, one of which will be described in section 2.5.5.

2.3.5 Cross Validation

Consider a specific split of data into a training, testing, and validation set. During training the model will fit to the underlying distribution, or eventually the noise with enough model complexity, however, a similar problem can be seen

with the validation set. If the validation set is used to determine the optimal hyperparameter choices, then it stands to reason that the choice of parameters is inherently dependent on the selection of the validation set. Since the set is predetermined randomly before the training process, the model will suffer from the variability in said selection, when selecting hyperparameters. Additionally, as detailed in 2.3.3, getting more data reduces the variance, which means that if there is already little available data, or the validation set is selected to be sufficiently large, unstable models would suffer from the reduced number of training samples.

A solution to minimize the impact of the variability from selection of the validation set is using a training method called cross validation, chapter 5.1 [25]. Consider figure 4, the entire dataset is split into training and testing in step A). Then in step B), during training, the dataset is split into five equally large portions and the model is trained five separate times. The selection of five is completely arbitrary, and is purely done as an illustrative example. In each training iteration, the validation data consists of one of the five components, different each time, and the training set is the rest of the samples. Similarly to how a model can incur variance from the choice of the validation set, each of the individual models that are trained can as well, however, since five models are trained, we can average their contributions and determine an average score for a particular model.

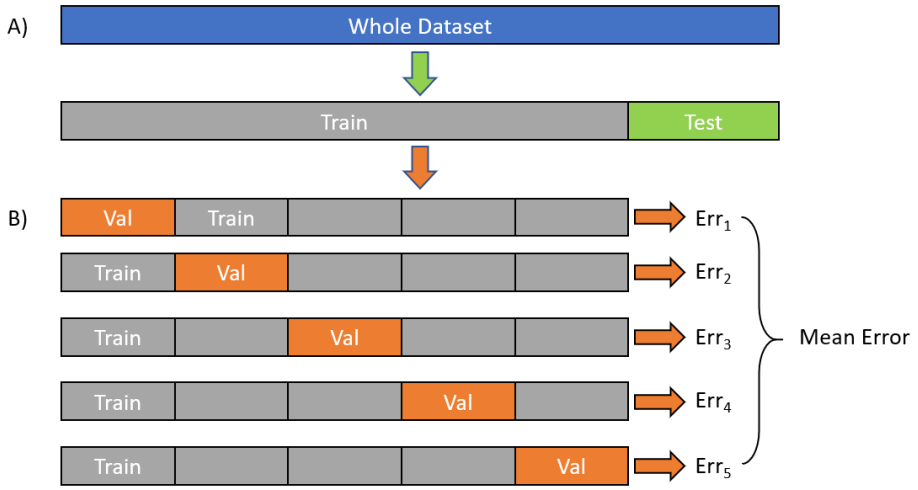

Figure 4: In step A) the whole dataset is separated into a test and training set, and in step B) cross validation is used to train a model. The training set is separated into k number of folds, in the image $k = 5$. Then the model is trained k times on different combinations of the folds with one fold held out for validation of said trained model.

A special case of the k-fold cross validation (k-CV) is leave one out (LOO)

cross validation, chapter 5.1 [25]. For LOO the choice of k is equal to the number of training samples, hence the training set consists of all $n-1$ samples and only a single sample is used for validation. The contribution of each model is combined to get an average performance, similarly to k-CV. LOO has the consequence of requiring a lot more computation, leading to it being unfeasible to use.

The two methods have an impact on the variance and bias of a model. In chapter 5.1 of An Introduction to Statistical Learning [25] it details that k-CV improve the variance over a simple predetermined split, or more specifically compared to a 50-50 split of the dataset into training and validation. Certainly in this case, the reduction in training samples have an impact on the inherent variance of the data that the model learns from. Comparing LOO and k-CV, the former is close to an unbiased estimate of the whole training set, given that the training set in each iteration of LOO consists of almost the entire training set, i.e. $n-1$ samples. Comparatively, k-CV contains a smaller proportion of the total training set for when $k < n$, leading to a more biased estimation of the total training set, however, still smaller than the bias inherent in a predetermined random split. When it comes to variance, they argue that the reduction in bias from using LOO over k-CV comes at the cost of increased variance. The increase in variance stems from the high positive correlation between the n trained models given the relatively small change removing one sample will have on the trained model. Furthermore, the mean of many highly correlated values have a high correlation, thus an increase in the variance of the model. k-CV still suffers from the positive correlation between models, however, to a lesser extent, thus incurring less variance.

That being said, other researches have a different view on the difference between LOO and k-CV. For instance [53] argues that simply stating that LOO suffers from more variance than k-CV is not entirely correct, given that it is dependent on the context of the use of cross validation. The paper details a number of experiments done for LOO, k-CV, and k-deletion cross validation, and shows that the increase in variance for LOO over the other methods occurs for models like LASSO (l1 weight decay loss for regression model) and SCAD (a non convex weight decay loss), see [14] for specifics of the SCAD loss function. This is due to the uncertainty incurred from small penalty coefficients and large feature space.

For the purpose of this thesis, cross validation will be used to select hyperparameters and to select between possible models, see section 3.6 for the specific hyperparameters spaces for the models that was used.

2.3.6 Randomized Grid Search over Hyperparameter Space

For models like KNN and polynomial regression a single hyperparameter needs to be selected, number of neighbours and degree of the regression, respectively, to use the algorithms. One approach would be to train a model for each possible choice within some reasonable set domain and select the best performing, according to some predefined metric, choice. This approach remains simple

enough when a model only has one possible parameter. However, for models with considerably more possible options can be quite resource intensive. An alternative approach is to do a randomized grid search over possible combinations of hyperparameters.

That is for some hyperparameter space $H = S_1 \times S_2 \times ... \times S_k$ consisting of sets S_i of some number of categorical options or continuous options, each iteration of the randomized search would select an element $h \in H$ where $h = (S_{1,i_1}, S_{2,i_2}, ..., S_{k,i_k})$ a vector of elements of each of the parameter spaces S_i. A model is trained, through the process of cross validation, and scored according to a predefined metric. The combination of parameters that perform the best is selected, or alternatively some specific parameter options can be eliminated and a more refined search can be performed again.

A randomized grid search allows for more complex hyperparameter spaces for models to be explored, and estimated, at a reduced computational cost. Additionally, even if a single model only has a single parameter to determine, the model could be used in conjunction with other models as a pipeline, leading to the total hyperparameter space being considerably higher. As detailed in section 3, the use of feature selectors and dimension reduction methods can be added, and explored, in a randomized grid search over the hyperparameter space.

2.3.7 The Curse of Dimensionality

One problem of particular importance to this thesis is high dimensional data, that is the feature space is larger than the sample space. High dimensional data posses a number of problems related to machine learning, the first of which is the locally sparse neighbours around each sample. Consider a point p in some metric space, then the epsilon neighbourhood of point p is the set $B_\epsilon(p)$ such that $B_\epsilon(p) = \{x \in X \mid d(p, x) < \epsilon\}$ for some metric d where X is the sample space. Given the space ordinarily used in machine learning is a euclidean space, the metric will be the euclidean metric $d(p, x) = |p - x|$ unless otherwise stated. As the dimensionality increases, the volume of the epsilon ball shrinks, past a certain point. To see this, consider the function of the volume of a unit hypersphere as the dimensionality increases. This is shown in figure 5. The formula for the volume of a hypersphere is $V_n = \frac{2\pi}{n} V_{n-2}$, thus when $n > 2\pi$ the volume decreases, as shown in the figure. This reduction in volume means the local neighbourhood gets more and more sparse as points get further and further apart, given that the volume of space the sphere encompasses decreases.

An alternative way of presenting this problem, courtesy of chapter 5.11 of [18], is to consider a categorization of each feature into ten unique values. When the number of features is only one, the total number of samples needed to have at least one sample of each unique value is quite small. When the dimensionality increases, the number of samples necessary to fully express all possible combinations increases exponentially with the dimension. In two dimensions there would need to be at least 100 samples, but in just six dimensions there would need to be at least one million samples. However, each unique combination

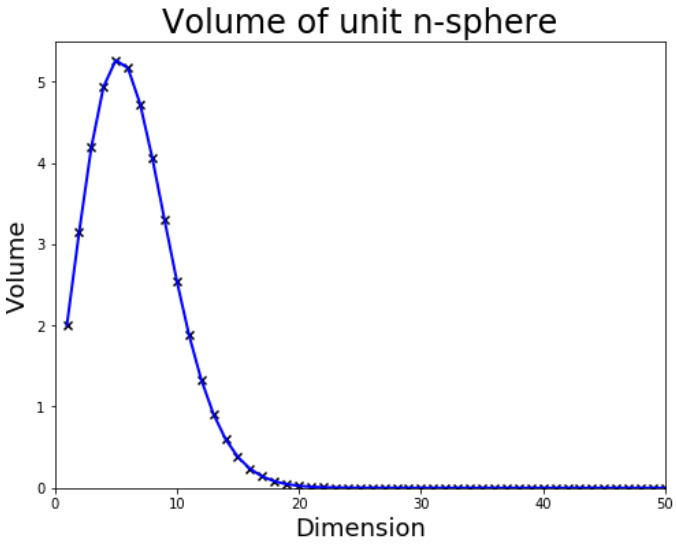

Figure 5: The figure shows the volume of a unit n-sphere against the number of dimensions. As the number of dimensions increases, the volume rises to a peak and then falls, tending to zero as $n \to \infty$. Mathematically the volume of a unit n-sphere is $V_n = \frac{2\pi}{n} V_{n-2}$ in relation to the volume of an (n-2)-sphere, thus the volume would decrease when $n > 2\pi$.

need not be relevant, however, without having a sufficient number of samples, a model would have no knowledge on whether the combinations it does have is sufficient or not.

The underlying problem is that distance metrics become less useful with more dimensions. Consider a set of independent and identically distributed data samples $X = \{x_1, ...x_n\}$ and a random reference point Q, then as the dimensionality increases, the minimum and maximum distance become indescribable. More formally;

$$\lim_{k \to \infty} P\left(\text{DMAX}_k \leq (1+\epsilon)\text{DMIN}_k\right), \tag{8}$$

where $\text{DMIN}_k = \min\{d_k(x, Q) \ \forall \ x \in X\}$, $\text{DMAX}_k = \max\{d_k(x, Q) \ \forall \ x \in X\}$, and epsilon is some arbitrary positive number $\epsilon > 0$, see [4]. This means that models that rely on a distance metric, or nearest neighbours, break down in higher dimensions under the assumption of the data being independent and identically distributed.

Given the nature of the data that is used in this thesis, as described in section 3.1, properly mitigating the difficulties posed by higher dimensional data is crucial. The next section will detail a way to mitigate the high dimensionality in the form of Feature Engineering.

2.3.8 Feature Engineering

Data that is considered high dimensional, i.e. $m > n$ where m is the number of features and n the number of samples, suffer from breakdown in the distance metric and local neighbourhood sparsity. Feature Engineering is one way to mitigate these problems by reducing the dimensionality of the data from m to some lower dimensional space $\hat{m} < m$, and usually $\hat{m} < n$ by some function $\phi : \mathbb{R}^m \to \mathbb{R}^{m'}$. There are two principle ways of reducing the dimensionality, the first is selecting a subset of features that is deemed most relevant for the given task or possess some important characteristic, or perform a linear, or non-linear, transformation of the feature space and embed it in a lower dimensional space. The latter being called dimension reduction or representation learning and the former being subset selection.

Dimension reduction methods involve some linear or non-linear transformation of the feature space to a lower dimensional space. A simple example of such a method is Principle Component Analysis (PCA), chapter 10.2 [25]. PCA creates a set of orthogonal basis vectors of the covariance matrix for the data samples. A subset of those basis vectors can be selected to perform dimension reduction. The selected basis vectors are based on the eigenvalues for those orthogonal basis vectors. Other more complex methods exists with different desirable properties. For instance, Universal Manifold Approximation (UMAP) uses a local distance metric between nearby points to find a lower dimensional embedding [37]. Dimension reduction methods serve an important role in feature engineering when the features are deemed important, however, if a feature is simple noise, then the noise will be included in the lower dimensional representation. Furthermore, to classify any new samples the whole original feature space is needed, something that might not be a desirable property. However, methods such as PCA and UMAP are good visualization tools for high dimensional data in two dimensions.

The form of feature engineering that will be used in this thesis is feature selection. There are a number of different ways to performing feature selection, but principally there are two approaches that will be used, filter methods and wrapper methods. A filter method is a simple statistical approach ranks each feature for its importance. In general, these methods are univariate approaches for determining relevant features. This means that filter methods tend to select variables that are highly correlated. Certain methods have been developed to counteract such behaviour, like Fast Correlation Based Filter (FCBF), however, they are outside of the scope of this thesis [51]. The types of filter methods used are described in more detail in section 2.4. Wrapper methods consist of training a classifier model on the data and using metrics on said model to determine relevant features, chapter 6.2 [25]. The simplest approach is applying l1 weight decay to a classifier and using the variable coefficients as a measure of a feature's importance. Other methods, like Random Forest [6], inherently rate each feature as part of the learning process, hence that can be used instead.

2.4 Filter Methods

Four types of filter methods will be used as part of this study, those being One-way Analysis of Variance (ANOVA), Fisher Score, ReliefF, and Mutual Information. Each of the methods will briefly be explained below.

ANOVA computes the F-statistic $F = \frac{(TSS-RSS)/p}{RSS(n-p-1)}$ where n is the number of samples, RSS is the sum of residual squares, TSS is the sum of total squares, and p is the number of relevant features for testing the null hypothesis $\beta_0 = \beta_1 = ... = \beta_p = 0$ of a regression model, see [25] chapter3.2. Given that the analysis is univarate, $p = 1$. ANOVA does require that the scale of each of the features are the same, which is guaranteed by the standardization done in the preprocessing step.

The Fisher score selects features such that distance to other classes is as large as possible and within class distance is small [19]. Specifically the Fisher score is $F(\mathbf{X}_j) = \frac{\sum_{k=1}^{c} n_k(\mu_{j,k}-\mu_j)^2}{\sigma_j^2}$ where $\mathbf{X}_j$ is the feature vector of feature j, c is the total number of classes, n_k is the number of samples with class k, $\mu_{j,k}$ is the mean of the j-th feature vector for samples with class k, and μ_j is the mean of feature vector j.

ReliefF is an extension of the Relief algorithm that iteratively updates each features importance based on the closest within class samples and closest different class samples to the given sample [48]. Specifically it selects a random sample R_i in step i, then finds the k nearest neighbours with the same class H_j and k nearest with a different class M_j, called nearest hits and misses, respectively. The importance of each feature A is updated by the following rule;

$$\begin{aligned} W_{i+1}(A) =& W_i(A) - \frac{1}{mk}\sum_{j=1}^{k} \text{diff}(A, R_i, H_j) \\ &+ \frac{1}{mk}\sum_{C \neq class(R_i)} \left(\frac{P(C)}{1 - P(class(R_i)} \sum_{j=1}^{k} \text{diff}(A, R_i, M_j) \right), \end{aligned}$$

where $\text{diff}(A, I_1, I_2)$ is a function defined on samples I_1 and I_2 that is 0 if the values of feature A are equal for the two samples, otherwise it is 1. The algorithm has an extension to missing values, however, that is not important for this thesis.

Lastly Mutual Information is the Kullback-Liebler divergence between the join probability distribution of two random varaibles with the product of said marginalized probability distributions. That is;

$$I(X, Y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(X_i, Y_j) \log \frac{p(X_i, Y_j)}{p(X_i)p(Y_j)},$$

for discrete variables, or in terms of entropy $I(X, Y) = H(X)+H(Y)-H(X, Y)$. Further details on mutual information can be found in [29].

2.5 Classification Models

The following sections will detail different classifiers are used as either feature selectors and/or classification, see section 3.

2.5.1 Support Vector Machine

Support Vector Machine (SVM) is a popular supervised learning method that aims to select a decision hyperplane that separates a binary classification problem. The hard margin version was introduced in a paper in 1992 [5] and the soft margin version was presented three years later in [8]. The hard margin approach consists of selecting a hyperplane such that any sample x_i has a class $y_i = \text{sign}\big(\mathbf{w}^T x_i + b\big)$. However, there are many such hyperplanes provided the data is linearly separable, in the case it is not linearly separable the soft margin approach needs to be used. SVM specifically selects the hyperplane such that the margin is maximized. The margin is the distance between the hyperplane and the parallel lines at the closest point of each class. Maximizing the distance is equivalent to minimizing $\frac{1}{2}||\mathbf{w}||^2$ subject to $y_i(\mathbf{w}^T x_i + b) \geq 1 \ \forall i = 1, ..., n$, also known as the primal formulation of SVM. The data points that lie on, and inside, the margin boundary are called the support vectors and fully define the decision boundary.

The above formulation is the hard margin approach, however, given that data is not linearly separable in general, the following derivation will detail the soft margin approach. Instead introduce a slack variable $\xi_i \geq 0$ for each data sample that is the amount of miss-classification by the separating hyperplane, which is equal to 0 in cases where data points are not misclassified. The primal formulation of SVM is instead;

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \quad \text{s.t. } y_i(\mathbf{w}^T x_i + b) \geq 1 - \xi_i, \tag{9}$$

where C is a hyperparameter that determines the degree to which samples a penalized for being misclassified. Large C converges to a hard margin. The minimization problem can be solved by Lagrangian optimization. Consider a set of constraint parameters α_i such that $0 \leq \alpha_i \leq C$, then the Lagrangian is;

$$L(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i\big(y_i(\mathbf{w}^T x_i + b) - (1 - \xi_i)\big), \tag{10}$$

which solved by differentiation to find a minimum is;

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \quad \text{given } \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{11}$$

called the dual formulation. The dual formulation can be solved in terms of α yielding the following weights $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i$.

The formulation of the model so far is linear. One way to add non linear approximation to the model is to add a non linear feature functions $\phi : \mathbb{R}^m \to \mathbb{R}^{m'}$ from the feature space m to some larger feature space m'. Consider a new data sample x and some feature function $\phi(x)$. The weights are defined as $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)$ and the decision function is defined as $\hat{y} = \text{sign}\big(b + \mathbf{w}^T \phi(x)\big)$, the main computational cost of the non linear transformation ϕ will be computing the inner product $\phi(x)^T \phi(x)$. However, with a specific choice of ϕ, the inner product can be defined as a function in the original feature space of size m, hence forgoing computing in the feature space of size m'. Any such choice of ϕ will lead to a kernel function $K(x, x') = \phi(x)^T \phi(x')$ and the decision function can be rewritten as $\hat{y} = \text{sign}\big(b + \sum_{i=1}^{n} \alpha_i k(x, x_i)\big)$. The kernels that will be relevant for this thesis are the Polynomial, Radial Basis Function (RBF), and Sigmoid kernels, each defined below:

$$\begin{aligned} \text{Polynomial: } K(X, Y) &= (\gamma X^T Y + r)^d, \\ \text{RBF: } K(X, Y) &= \exp(-\gamma ||X - Y||^2), \\ \text{Sigmoid: } K(X, Y) &= \tanh(\gamma X^T Y + r), \end{aligned}$$

for some hyperparameters γ, r, and d.

2.5.2 Logistic Regression

Logistic Regression was briefly introduced in section 2.3.2. For any given sample x, logistic regression assigns a probability $P(y = 1|x) = \sigma(\mathbf{w}^T x + b)$. The decision boundary is $P(y = 1|x) > 0.5$, or simply the decision hyperplane $\mathbf{w}^T x + b = 0$. The optimal choice of weights $\mathbf{w}$ is the maximum log likelihood, see equation 3. The exact method of optimization for logistic varies. For the purposes of the implementation used in this thesis, liblinear [15] and lbfgs [33] solvers are used, as detailed in the Sci-kit learn documentation, see software section 3.2, for logistic regression. The elastic-net logistic regression model is optimized by stochastic gradient descent instead of liblinear or lbfgs.

2.5.3 Stochastic Gradient Descent With Modified Huber Loss

It is a bit of a misnomer to have Stochastic Gradient Descent (SGD) as a classifier, given that SGD is an optimization algorithm, however, the references to a SGD classifier in this thesis is specifically SGD used to optimize Modified Huber loss [52]. Specifically the model optimizes the total loss $\frac{1}{n} \sum_{i=1}^{n} \big(L(y_i, f(x_i))\big) + \lambda R(\mathbf{w})$ where $R(\mathbf{w})$ is the weight decay regularization and $L(y_i, f(x_i))$ is the modified Huber loss function;

$$L(y_i, f(x_i)) = \begin{cases} \max(0, 1 - y_i f(x_i))^2 & y_i f(x_i) > 1 \\ -4 y_i f(x_i) & \text{otherwise} \end{cases} \tag{12}$$

where $f(x) = \mathbf{w}^T x + b$. This loss function is less sensitive to outliers than ordinary least square loss. This is but one of many options for loss functions,

however, given linear SVM and logistic regression is used as part of the feature selection to begin with, only modified Huber loss will be used.

The process of optimization of SGD involves iteratively updating the weight parameters by the gradient of the weights until it reaches a convergence or a maximum number of iterations. Specifically the t-th iteration, the weight update rule would be;

$$\mathbf{w}_{t+1} = \mathbf{w}_i - \gamma\bigg(\lambda\frac{\partial}{\partial\mathbf{w}}R(\mathbf{w}) + \frac{\partial}{\partial(\mathbf{w})}L(y_i, f(x_i))\bigg), \tag{13}$$

for some learning rate γ. The choice of γ is selected as "optimal" in the Sci-kit learn package, which equates to $\gamma_t = \frac{1}{\alpha(t_0+t)}$ at time step t and a heuristic parameter t_0. This is an example of a decaying learning rate, that is the proportion of change is reduced as learning progresses.

2.5.4 Decision Tree

A Decision Tree is a non-parametric model that can be used for classification or regression, however, for this thesis I focus on classification. The model consists of a set of simple decision rules based on features of the data, starting at the root and ending at leaf nodes. At each node a single feature is selected, based on an information metric, and a specific split is selected to determine the decision function for its two children nodes. The process of building new nodes continues until it reaches a specified maximum depth or all samples at a given node consists of a single class, hence the node will be a leaf node. This means that the tree representation of a Decision Tree need not be balanced. The type of information measure used to select a feature is a set parameter, either Entropy $H(Z) = -\sum_c p_c + \log p_c$ and Gini Index $G(Z) = \sum_c p_c(1-p_c)$ for some subset Z of the dataset X and proportion of presence, also refereed to as the class probability, p_c for class x.

Note that Decision Trees tend to overfit heavily when the data is high dimensional, and its use for feature selection is simply to see the possible features it would use, more so than an assumption that the model could select very good predictive features.

2.5.5 Random Forest

Random Forest is an ensemble method that trains many smaller estimators and combines their predictions to make one final prediction for each sample [6]. Each estimator is a small decision tree estimator that gets a random subset of the available features to consider, thus reducing the bias the model can have for any given feature.

Given the nature of Random Forest classifier, it will naturally select a lot of relevant features, even when few features are the only relevant ones. This means that feature selection on the basis of a Random Forest model selects very many features at relatively similar feature importance. Furthermore, the feature importance tends to favour features that have many unique values.

2.5.6 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a simple classification algorithm that predicts a class c for a point p_i where c is the majority class in the k closest points to p_i based on a given distance metric, chapter 7.2 [36]. KNN is an example of a non parametric model that can fit unrealistic decision boundaries for a low number k. In addition to selecting k, the model is also dependent on the choice of distance metric. For the purposes of this thesis the distance metric will be Minkowski distance of degree s, either 1, 2, or 3. Minkowski distance is defined as $d_s(x, y) = \left(\sum_i |x - y|^s\right)^{1/s}$. Euclidean is a special case when $s = 2$.

2.5.7 Gaussian Naive Bayes Classifier

The Naive Bayes classifier [41] is built on the relationship between the posterior and likelihood with an assumption of conditional feature independence. Consider a Bayesian inference problem;

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)},$$

for some model parameters θ and data D. The term $P(\theta \mid D)$ is the posterior probability of parameter θ given the input data, $P(D \mid \theta)$ is the likelihood of the data given the model parameter, and $P(\theta)$ is the prior probability of the model parameters before observing any data. For classification the goal is to predict class y_i based on the data samples $x_{i,1}, ..., x_{i,m}$. Consider a single data sample labeled E and the predicted class of said data sample y for simplicity. The features of E are then $x_1, ...x_d$.

The conditional feature independence assumption means that it is assumed that the features are independent of each given the class;

$$P(x_j \mid y, x_1, ..., x_{j-1}, x_{j+1}, ..., x_m) = P(x_j \mid C),$$

that is the likelihood can be simplified as the following product via marginalization $P(x_1, ..., x_m \mid y) = \prod_{j=1}^{n} P(x_j \mid y)$. Given this simplification, the Naive Bayes classifier assigns class y to a sample E that has the highest posterior probability;

$$\hat{y} = \arg\max_y P(y) \prod_{j=1}^{n} P(x_j \mid y).$$

The choice of using the naive assumption leads to poorly calibrated probabilities, however, the predicted class, based on the maximum posterior, is often correct.

The specific type of model used in this thesis is the Gaussian Naive Bayes classifier, where it is assumed that the likelihood probability is Gaussian. That is $P(x \mid y) = \mathcal{N}(\mu_y, \sigma_y)$ for some mean μ_y and and standard deviation σ_y for class y.

3 Methodology

3.1 Datasets

The data used for this study comes from two data sources, The Cancer Genome Atlas (TCGA) dataset for colon adenocarcinoma (COAD) and Haraldsplass Diakonale Hospital (HDH, or HDS in Norwegian) Colon Cancer (CC), denoted as TCGA and HDS, respectively. For each data source, this study will use both the mRNA and miRNA raw count signatures for the relevant cohorts. The four distinctive datasets will thus be refereed to as TCGA mRNA, TCGA miRNA, HDS mRNA, and HDS miRNA. In addition to the raw counts there are clinical parameters for each datasets; age, sex, recurrence, overall survival, mismatch repair status (MMR), and more, that will be used in conjunction with the raw counts.

Specifically for the TCGA source, there are a number of missing values pertaining to MMR and microsatellite instability (MSI) status, among others, leading to a poor comparison between the two. For the purposes of this study, the MMR status of the HDS cohort will be compared with the MSI status of the TCGA cohort, given the similar dynamics of the underlying mechanism, and the fact that the majority of TCGA cohorts have a MSI status. Similarly HDS defines the recurrence for each patient with a cutoff of 5 years, due to the study limitation, while TCGA has a 12 year study limitation, and defines instead the progression free interval (PFI), which should be comparable to the recurrence parameter of HDS. All patients that have missing information for critically important parameters as part of the study will be excluded when relevant, see further discussion when incorporating analysis on the clinical data.

The exact specifications of how the miRNA HDS dataset was recorded can be found in [24]. The pipeline for mRNA has a similar set of primary procedures. Briefly summarized, samples of the fresh frozen tumor was extracted using miRNeasy, Mini Kit, and homogenized with Tissuelyzer. The mixture is purified by DNase treatment and the RNA concentrations are measured using NanoDrop and quality by Agilent RNA Bioanalyzer. Until the samples were used, they were stored at $-80°$C.

From those samples, the miRNA sequences was converted to a FASTQ format and the expressions were read using miRDeep2. The mRNA counts were read using Illumina TruSeq Stranded Total RNA. The reads were aligned with the human genome GRCh38.p10 using hisat2 and Gencode transcriptome reference release 26, which was processed with Samtools and FeatureCounts. The study focused on protein encoding mRNAs. The HDS study was approved by the Regional ethics Committee according to the Helsinki Declaration.

The pipeline for collecting TCGA samples is outlined here for miRNA and here for mRNA. This only outlines the pipeline post FASTQ file format for read alignment.

	HDS		TCGA	
	miRNA	mRNA	miRNA	mRNA
Number of Patients	128	79	416	446
Mean Age	72.05	71.91	67.57	67.73
Relapse True	45 (35%)	20 (25%)	106 (25%)	112 (25%)
Relapse False	83 (65%)	59 (75%)	306 (75%)	334 (75%)
TNM Missing	-	-	11 (3%)	11 (2%)
TNM 1	18 (14%)	2 (2%)	71 (17%)	75 (17%)
TNM 2	54 (42%)	43 (55%)	159 (38%)	177 (40%)
TNM 3	36 (28%)	34 (43%)	114 (27%)	122 (27%)
TNM 4	20 (16%)	-	61 (15%)	61 (14%)
Features Pre Filter	2588	19817	2155	20531
Features Post Filter	900	14261	1206	14427
Feature Intersection	768 (85%)	12021 (84%)	768 (64%)	12021 (83%)

Table 2: Table showing the distribution of clinical information for each of the four datasets and the size of the feature space (gene pool). The patient numbers are all patients with a tumor sample after having filtered away patients with missing Relapse/PFI value. For HDS there are more patients with miRNA than for mRNA, hence the different in cohort size, while for TCGA it is reversed. For the purpose of relapse, note that TCGA does not have a relapse parameter, hence in this table PFI is used synonymously with Relapse. In TCGA there are a number of features that are missing, and of the cohort used, a small percentage have no TNM-stage. For the purpose of the analysis these patients will be included and excluded depending on the specified selected cohort subset. The last three rows detail the shape of the feature space. The first of the three show how many features are prior to library size filtering is performed. The second is the total number of features after said filter. The last is the intersection of the feature space between the two data sources. The number is the total number of features, while the percentage is how many of the features of a data source is in the intersection.

The distribution of clinical data and feature space for each dataset is shown in table 2. The HDS dataset have considerably fewer samples than the TCGA datasets. This means learned models on the HDS dataset have a higher risk of overfitting if the model complexity remains the same on both sets. The mHDS dataset have considerably fewer stage 1 and 4 patients than the other datasets, hence why comparison between mTCGA and mHDS would be more comparable on just stage 2 and 3 samples, see section 3.3 for details on subset selection for the experiments. There is also a considerable difference in mean age between the two data sources, 72 in HDS to 67.7 in TCGA. Given that colon cancer patients are in general old and live for a relatively long time after surgery, the impact of other factors on a patients help can be relevant for overall survival analysis. The thesis focuses on the relapse survival, instead of overall survival, but even so, the relevance of mean age will be discussed in section 4.1. Furthermore, only about 25% of samples have relapse, hence the dataset has a $25 - 75$ class imbalance,

although the miHDS dataset does have slightly more relapse samples. The class imbalance problem will be briefly touched on in section 3.5.

Lastly, of note, is the difference in the total feature space between the datasets. The prefiltering, discussed in section 3.4, remove significantly more features for the miRNA dataset compared to that of the mRNA datasets. This is even more impactful given that the original feature space is about one tenth the size. The small feature space in the miRNA intersection between the two data sources will be briefly touched on in section 3.3.

3.2 Software

The processing of the raw gene counts was done in R using two packages edgeR [42] and DEseq2 [34]. The former being used for gene expression filtering and the latter for normalization procedure.

All other code was done in Python3 with a number of relevant packages. Numpy and Pandas were used to organize the data and perform vectorized operations for data handling. Matplotlib and Seaborn were used to generate all the graph figures shown in this thesis. Sci-kit learn [39] was used for almost all machine learning methods used, those being SVM, Random Forest, Gaussian Naive Bayes, KNN, Decision Tree Classifier, Logistic Regression, and SGD. Additionally, the Mutual Information and one-way ANOVA analysis was performed using Sci-kit learn functions. ReliefF and Fisher was used from the skfeature module [32] (note that a separate branch of the module called chapper-skfeature was used) and the attempts at using the Genetic Algorithm to perform subspace search, see section 4.8, was based on mealpy [50]. Some models, see section 3.3, was retrained with a random over sampler. The module used for this oversampling was imbalanced-learn [30]. Lastly, survival analysis was done using the Lifelines module [11].

3.3 Method Overview

The main problem is finding a subset of features that can be used to train a classifier. The approach of this thesis is to use a number of different feature selection methods, either wrapper methods training specific classifiers that rank each feature or filter methods that filter away features based on statistical information about the feature values. For each of the feature selection methods a set of features will be selected. However, models could find different number of relevant features, and models with more features will in general lead to higher training and validation score, but could lead to poorer generalization scores, thus the top k number of features from each selector is chosen and experiments are run on each selection of k and compared. The choices for k are $[2, 3, 5, 7, 10, 25, 50, 100, 150, 250]$. It should be noted that the figures shown in section 4 will not show the case for 250 features, given that the trend is apparent without including 250. Additionally, this means the figures will be of a 3×3 grid of subplots, making them more readable.

Each of the feature selection methods, with a specified number of features k, is combined into a pool of different feature sets of size k. This pool of different feature selection methods is used as a hyperparameter in a randomized grid search for each of the four classification algorithms that are trained, those being Support Vector Machine, Random Forest, k-Nearest Neighbours, and Gaussian Naive Bayes classifier. This means that there is one randomized grid search for each combination of k and model of those four mentioned above. This exact pipeline can be seen in figure 6. Note that prior to performing feature selection and/or randomized grid searches, the original counts go through the normalization procedure that is explained in section 3.4.

The whole pipeline, as outlined in figure 6, is done for each of the datasets separately. The models that are trained on one data source is then tested on the other data source to determine if the model generalizes well. Note that to do this, the feature space has to be the intersection of the feature space of the two data sources, hence some features are discarded when performing this analysis. Furthermore, given the different distribution of TNM-stage for the two data sources, the comparison between the two data sources is primarily done when looking at only stage 2 and 3 patients. However, all experiments on mRNA and/or miRNA data uses the intersection of gene pool between the two data sources, that way a model can be tested on the other data source. Specifically, the following list details each type of experiment that is done;

- (A) Stage 2 + 3: this experiment is done only on TNM-stage 2 and 3 patients.
- (B) All Stages: this experiment is done with all samples, regardless of TNM-stage.
- (C) Clinical: this experiment is only done on the clinical information from the two data sources. Only age, Tumor Stage, Node Stage, Metastasis Stage, and MMR deficiency are used. Note also that for purely clinical study, the data used is the highest sample size dataset from each source, that means comparing miRNA HDS clinical data to mRNA TCGA data. However, the nature of what dataset the clinical data is collected from has no importance.
- (D) Stage 2 + 3 Transcriptiom: this experiment is done only on TNM-stage 2 and 3 samples where the total gene pool from mRNA and miRNA was combined prior to feature selection.
- (E) Stage 2 + 3 Transciptome + Clinical Data: this experiment combines clinical parameters with the mRNA or miRNA data post feature selection for TNM-stage 2 and 3 samples.

Each experiment is trained on one data source and tested on the other data source to determine if the model has generalized. However, given a specific combination of hyperparameters selected by the randomized grid search, the models

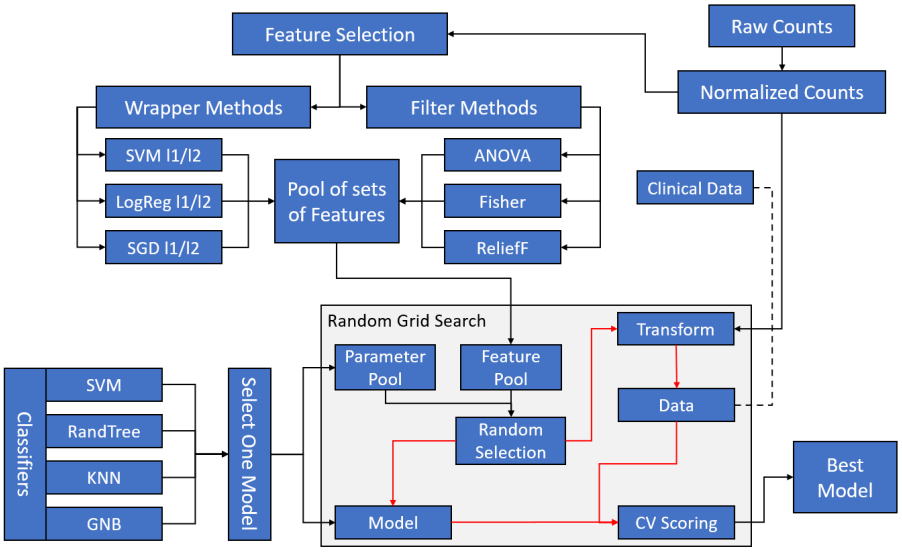

Figure 6: The figure shows the pipeline for model selection. Raw counts are normalized, then used in a feature selection scheme. A number of models is trained, or filter methods used, to select a number of relevant features. These sets of features are stored and used as an additional hyperparameter in the randomized grid search pipeline. The pipeline for the grid search consists of a dimension reduction step that selects reduces the gene space of the data to that of one randomly selected feature set (for instance it can select the SVM l1 set of features or ANOVA or some other) and one model with a random selection of hyperparameters based on the models hyperparameter space. The model is trained and scored through cross validation, and the best scoring models for each classifier is selected. The red lines indicate the iterative process of the randomized grid search. The line from the clinical data indicates that the transformed normalized counts is concatenated together with clinical parameters, thus the data used is a combination of genes and clinical information. However, the model is also trained without the addition of the clinical data, hence why the line is dotted. Do note that the feature selectors select the k best features, however, the number k is determined prior to the randomized grid search. This means a grid search is performed for each combination of classifier model and number of genes k, where k is selected from a set of predetermined numbers.

are also trained and validated on its own data source. This gives an overestimate of the validation score of that data source due to the whole dataset being used to determine parameters for the model. The reason for doing so will be outlined in section 4. This means that for each experiment (five different ones) all four classification models are optimized in their respective hyperparameter space for each of the 10 different number of features selected from the different

feature selection methods. This leads to a high number of possible models that have been tested, which will be discussed in detail in section 4.

3.4 Preprocessing of Data

The raw counts of the mRNA and miRNA expression has to be normalized based on its library size, that is the relative magnitude of each gene for a given patient. This is done through the DEseq2 package as part of R. However, prior to this normalization step, genes that have a low count will be filtered out, given that there is an uncertainty in the number of counts.

The filtering used is a function called filterByExpr in edgeR. Genes are kept if their count-per-million (CPM) are above a threshold for a proportion of the samples of the smallest sample group and have above a certain minimum total count threshold. Specifically the minimum count of 70% of the samples must be 10 and the total number counts must be above 15 for all samples. The function allows for filtering based on the response target, however, it was decided to not filter based on the response, given that it would bias the filtering on the response. The principle idea of the filtering is to instead filter away genes that are too lowly expressed to be considered biological and statistically significant.

The normalization process is done via the DNSeq2, see [34], library using a Variation Stabilizing Transformation (VST). The VST algorithm aims to find a differential function $h(X)$ such that the variance of the first degree Taylor expansion of h is approximately constant. This amounts to integrating $h(y) = \int^y \frac{1}{\sqrt{v(u)}} du$ for expectation u and variance $v(u)$ for some randomly distributed data. This amounts to transforming the data by $h(y_i) = \frac{y_i + a_i}{b_i}$ for some parameters a_i and b_i that can be estimated by maximum likelihood. The full details are described in [22].

After normalizing via VST, the data is then standardized per feature by subtracting the mean and dividing by the standard deviation. This standardization is done primarily because it improves the performance of a number of machine learning models, but also because it allows for easier comparison of relevant coefficients for feature selection. Additionally, it allows for the SVM classifier to not stall when using a polynomial kernel function. Given that features with very little variation prior to this standardization technique can show up has having more variation post standardization, features that have a low variance should be filtered out. This, however, is not an issue given that the output of the VST algorithm has significant variance for all features, hence a variance filter would not remove any features.

3.5 Methods for Feature Selection

Feature selection was performed using the filter methods described in section 2.4 and wrapper methods using some of the classifiers described in section 2.5. The filter methods was specifically used to select features based on its score, not its associated p-value for the null hypothesis. The latter is not used given

the p-value adjustments selected very few, if any, features, and the main goal is to analyse the performance with different number of selected features.

For the wrapper method, the following classification models were used as feature selectors;

- Linear SVM: the linear SVM model was used to extract features with both l1 and l2 loss. The set of features will be referred to as SVM_l1 and SVM_l2.
- Logistic Regression: the logistic regression model was used to extract features with l1, l2, and elastic-net loss. The models will be referred to as Log_l1, Log_l2, and Log_net. Note that the Log_net model was trained using stochastic gradient descent, while l1 was with liblinear and l2 was with lbfgs solvers.
- Stochastic Gradient Descent with Modified Huber Loss: all of l1, l2, and elastic-net loss was used to extract features with this model. The model is referred to as SGD_l1, SGD_l2, and SGD_net, however, the use of SGD specifically means with Modified Huber Loss, given that SGD is simply a optimization algorithm and not a classifier.
- Random Forest: Random Forest is a ensemble algorithm consisting of many small decision trees. The feature selection model is referred to as Rand.

The selected features were based on the coefficients of the model for the given feature, or in the case of Random Forest, the feature importance. There are different ways of selecting what features are important, however, for this study I select the features that have the highest absolute feature importance or coefficient value. There is some concern with selecting features based on the magnitude of its coefficients, given features with different magnitudes would have different magnitudes for its coefficients. However, each feature is standardized to have feature mean zero and standard deviation one, hence no difference of scale. Given the nature of Random Forest classifier, it will naturally select a lot of relevant features, even when few features are the only relevant ones. This means that feature selection on the basis of a Random Forest model selects many features at relatively similar feature importance. Furthermore, the feature importance tends to favour features that have many unique values.

Additionally, models that are used to for feature selection was trained with random over sampling of the lower represented response. This was primarily only relevant for the SGD_net and Log_l2 models as the models initially struggled with a class imbalance.

3.6 Hyperparameter Space for Classification Models

The following four methods are used as classifiers; soft margin SVM, Random Forest, KNN, and Gaussian Naive Bayes classifier. Each of the classifiers have a set of hyperparameters that is searched over during the randomized grid search

with the exception of the Gaussian Naive Bayes classifier. Below is a list of the possible hyperparameter spaces for each of the three models with hyperparameters and a brief explained of the impact of said hyperparameter.

For SVM there are a number of hyperparameters that have an impact on learning. For this thesis, the hyperparameter space is listed below;

- C: the C parameter determine how hard the margin is. Possible options are 10^d for d in $[-5, -4, -3, -2, -1, 0, 1, 2, 3]$.
- Kernel: Kernels introduce non linearity to the otherwise linear estimator. Possible options are Polynomial, Radial Basis Function (RBF), and Sigmoid.
- Degree (only relevant for polynomial kernel): is the degree of the polynomial kernel. Possible options are 2, 3, 4, and 5.
- Gamma: Gamma determine the relative importance of each datapoint contribution. The possible options are scale $\frac{1}{m\text{Var}(X)}$, auto $\frac{1}{m}$, and 10^d for d in $[-4, -3, -2]$.
- Class Weight: can either be balanced and none, the former putting weight on samples according to its class distribution.

The C parameter determines how soft the margin is, a higher value meaning a more hard margin. The Kernel parameter changes what kernel is used for non-linear transformation, and the Degree parameter is the degree of non linearity for a polynomial kernel. The Class Weight parameter determines if the algorithm compensates for class imbalance or not. Gamma determines the coefficient for the kernel function. For instance for RBF kernel it would be $k(x, x_i) = \exp(-\gamma||x - x_i||^2)$.

Secondly, the Random Forest model have the following hyperparameters;

- Impurity Measure: can be either Gini Index or Entropy. Used to determine selected feature and splitting point for said feature in the Decision Tree estimators.
- Maximum Depth: the maximum cut off depth of each estimator. The possible options are none, 1, 2, 5, 8, and 12.
- Number of Estimators: total number of estimators used as part of the model. Possible options are 2, 5, 10, 25, 50, 100, 250, and 500.
- Class Weight: can either be balanced and none, the former putting weight on samples according to its class distribution.

The maximum depth parameter determines how shallow each of the estimators are and the number of estimators determine how many such estimators is used. The impurity measure parameter determines which measure is used to determine the feature to split, and where to split said feature. Lastly the class

weight parameter determines if the algorithm compensates for class imbalance or not.

Lastly, the following hyperparameters are relevant for the KNN model, those being;

- Number of Nearest Neighbours: the number of nearest points to determine a classification of a point. The possible options are 1, 2, 5, 7, 10, and 15.
- P: is the power of Minkowski distance, the possible options are 1, 2, and 3.
- Weights: determine the weight each neighbour has when determining a classification. Possible options are uniform and distance, the former being equal weight for all neighbours and the latter having the weight be the inverse of distance to said neighbour.

4 Results

Given the number of experiments conducted and the total number of models that was trained as part of those experiments, only select subsets of models will be shown in this section. Section 4.1 will detail the results for Clinical data for all samples from both TCGA and HDS, and problems related to how models use certain features. Next section 4.2 and 4.3 will detail the results for Stage 2 + 3 mRNA and miRNA data and the poor generalization. Possible explanation of said poor generalization is discussed in section 4.4 in addition to inspection of specific genes class data distributions. Transcriptome and Clinical data will be briefly touched on in section 4.6. The details on mRNA and miRNA for all stages will be presented in section 4.7, and lastly other possible approaches that was tried and abandoned will be detailed in section 4.8.

The primary goal is to train models on one data source and test it on the independently sampled other data source to determine if the models generalize well. Given that there are four possible combinations of training on a data set and testing it on the other data source

4.1 Clinical Data, Experiment (C)

The clinical variables are MMR status (MSI for TCGA data since there is too many missing labels for MMR), Age, Tumor stage, Node stage, and Metastasis stage. The Tumor and Node stage are converted to binary feature vectors via a OneHotEncoding, leading to a total of eleven features. Figure 7 shows the mean validation ROC of trained models and the test ROC on the other data source, the former in dotted lines and the latter in solid lines, for all TNM-stages. Note that the data used for the figure was mTCGA (n=446) and miHDS (n=128) clinical data for all cancer stages.

Of the models presented in figure 7, only the SVM models are poorly trained, reaching the baseline balanced accuracy of 50% despite their ROC curve. This

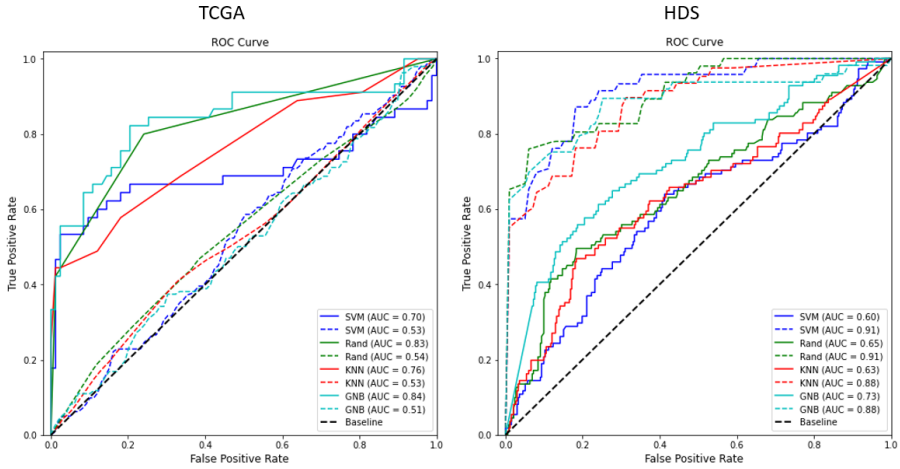

Figure 7: The figure shows the mean validation ROC, as a dotted line, and the test ROC, as a solid line, for both data sources side by side based on Tumor stage, Node stage, Metastasis stage, Age, and MMR status (experiment (C)). The training and testing is done on separate data sources. Models trained on TCGA perform better on HDS than during training on HDS and testing on TCGA. This is because of its reliance on the Age parameter, which has a reverse relapse signal in HDS compared to TCGA. Additionally, the TCGA trained models reach about 70% balanced accuracy, except the SVM model getting only a 50% balanced accuracy, which is equivalent to only predicting relapse for all samples. The HDS trained models are between 55 and 66%.

indicates the decision boundary is poorly calibrated and could achieve a better accuracy if it was tuned better. This is because the ROC curve is based on an optimal thresholding of probabilities, and given the default probability cutoff of 0.5, the accuracy was markedly lower even if the ROC curve look reasonable. Furthermore, the model fails to generalize regardless, see section 4.4 on a further discussion of the differences between the data sources. It should also be noted that all models are poorly trained on TCGA data, indicated by the dotted ROC curves near the line of chance. Thus, the model has not been able to learn anything meaningful from the TCGA data, struggling with the class imbalance problem.

Take for instance the Random Forest classifier trained on TCGA and tested on HDS. Despite having really poor validation error it still reaches an AUC score of 0.83 and a balanced accuracy score of 70%. Figure 8 gives an indication as to why. The Random Forest algorithm prefers values that have many unique values (high cardinality), and Age is the only variable with cardinality higher than 2. As seen in the figure the Random Forest algorithm has a relatively high feature importance on age, which would explain why it can generalize well on HDS. The age feature is standardized, i.e. mean zero and standard deviation

	Random Forest				LinearSVM			
	mTCGA1	mTCGA2	mHDS1	mHDS2	mTCGA1	mHDS1	mTCGA2	mHDS2
Age	nan	0.256	nan	0.491	nan	nan	0.035	0.036
MMR	0.188	0.092	0.145	0.113	0.116	0.297	0.109	0.306
N1	0.110	0.104	0.093	0.101	-0.000	0.443	0.000	0.447
N2	0.334	0.240	0.031	0.011	0.358	-0.005	0.361	0.000
N3	0.000	0.000	0.187	0.049	0.000	1.582	0.000	1.623
N4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
T1	0.000	0.000	0.013	0.000	0.000	0.208	0.000	0.175
T2	0.065	0.064	0.047	0.017	-0.114	-1.266	-0.104	-1.278
T3	0.160	0.144	0.219	0.148	-0.271	-1.045	-0.274	-1.054
T4	0.142	0.100	0.266	0.070	0.000	0.305	0.000	0.303

Figure 8: The table shows the feature importance and coefficient for each feature trained on mTCGA and mHDS clinical data for a Random Forest and LinearSVM classifier, respectively. mTCGA1 means that the model is trained without age as a feature, while mTCGA2 means it is trained with age as a feature. The Random Forest classifier puts a high importance on age over other features, which is to be expected given the feature has the highest cardinality out of the feature space. As a matter of fact, all other features have a cardinality of 2 given they are binary values. The LinearSVM model does not appear to put equally high importance on age given its coefficients.

one, however, the relapse age distribution might be different between the two data sources. Figure 9 shows that the relapse signal is reversed in HDS from TCGA, thus supporting the idea that the model is performing better because of a difference in the distribution of the cohorts, rather than it necessarily being the strength of the given feature. Furthermore, the LinearSVM coefficients indicate other parameters are more important than age, supporting the claim that the Random Forest classifier selected Age more due to the higher cardinality than it being a relevant feature. Given that it is only the SVM model that fails of the four models, it is also possible that Age is a important variable used by the other two models as well. The important point is Age is the only continues variable in the clinical data that is used.

Lastly it should be noted that when the analysis is performed only on stage 2 and 3 samples the performance is significantly worse than when using all samples.

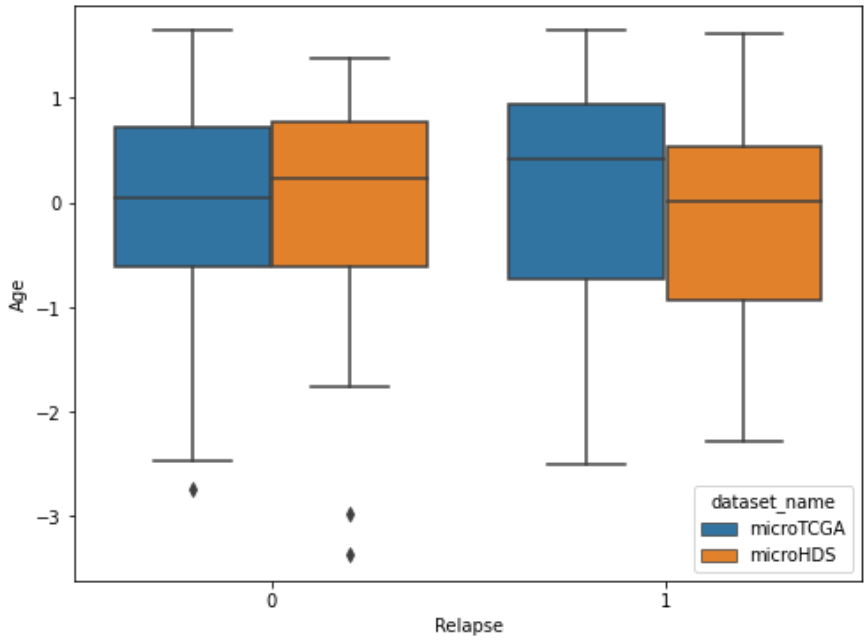

Figure 9: The figure shows a boxplot of the relapse distribution of normalized Age clinical feature for both data sources.

4.2 mRNA Stage 2 + 3, Experiment (A)

Consider the mRNA experiment A, as detailed in section 3, involves the intersection of gene pool across the two data sources for only stage 2 and 3 patients. The ROC curves for each model for different number of selected features trained on mTCGA and testing on mHDS is shown in figure 10. The dotted lines are the mean validation ROC for the trained models on mTCGA and the solid lines are the test ROC for those models on mHDS. Note that this specific experiment has the highest number of training samples of all four possibilities for experiment A, being 299 samples in mTCGA, but equally the smallest test set of 77 samples in the mHDS dataset.

The figure shows, quite naturally so, that the cross validation performance of the model increases with more features, however, only the SVM model takes full advantage of all the selected features as the number of features increases, reaching a near perfect ROC at 150 features, which shows clear signs of overfitting. This can clearly be identified as overfitting given the model has fully fitted to the underlying training data distribution yet fails to generalize to the other data source. The poor generalization is not only worse models above the line of chance, but also models performing well below the line of chance, indicating that the signal in the test set is different from the training set. A more detailed

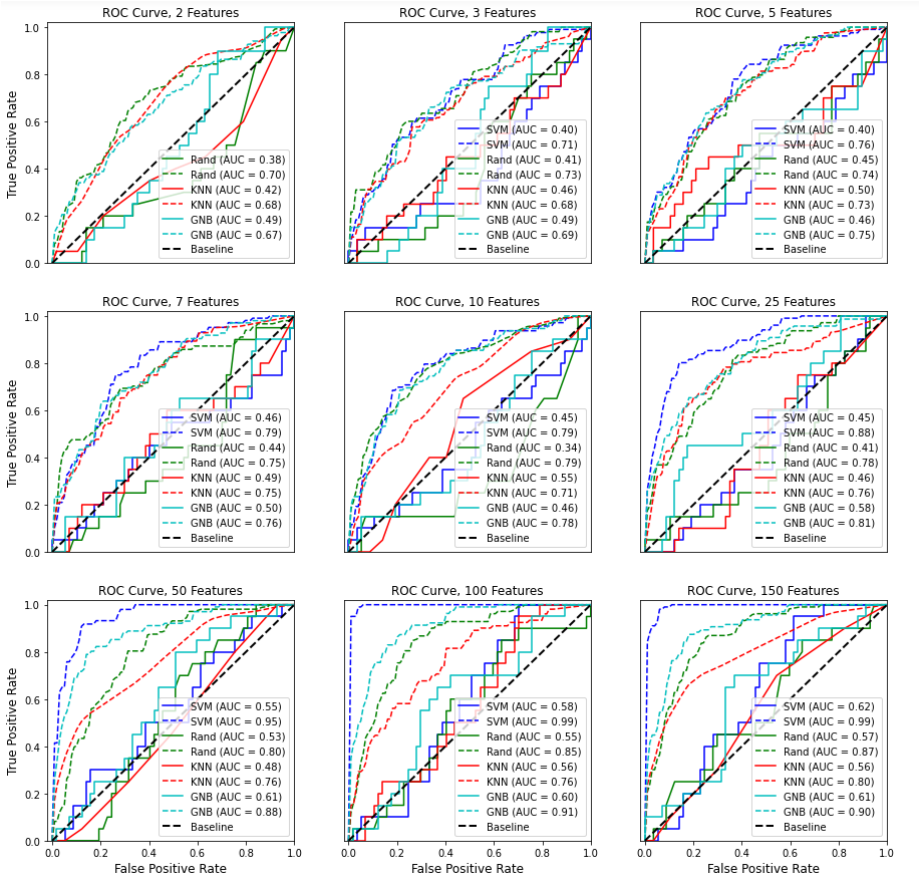

Figure 10: The figure shows the mean validation ROC on the mTCGA dataset and the test ROC curves on the mHDS dataset (experiment (A)). Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the mHDS dataset. The plots show a clear failure to generalize from learning on TCGA to predicting on HDS. Furthermore, some of the test ROC curves are below the line of chance, indicating that whatever signal was found was found to be reversed in the HDS dataset.

look at this observation will be discussed in section 4.4.

Beyond the SVM model, the KNN model stalls in cross validation ROC early at around seven features. Similarly the Random Forest classifier also does not markedly improve above ten features. It is difficult to discern exactly why the performance stalls. One possible explentation is that the feature selector used for that model is poor, however, for 100 features all of SVM, KNN, and GNB

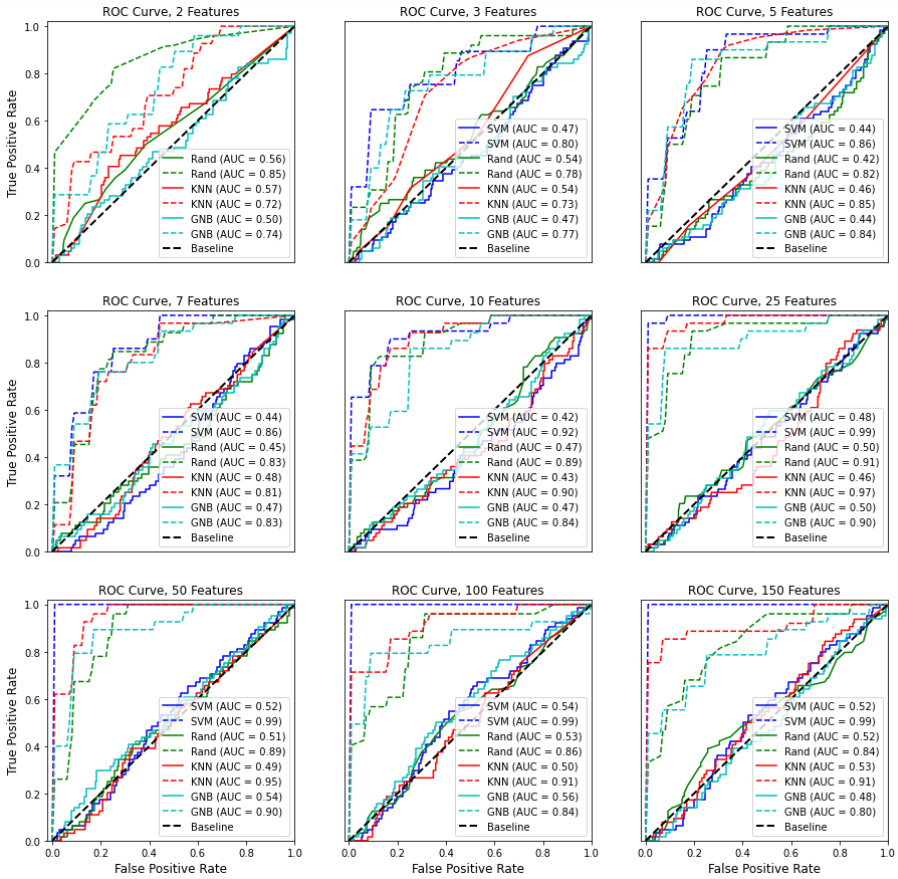

Figure 11: The figure shows the mean validation ROC on the mHDS dataset and the test ROC curves on the mTCGA dataset (experiment (A)). Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the mTCGA dataset.

selected the same feature selector, namely SVM_l1. Yet only the KNN model fails to improve based on that feature selector. It is possible there is an issue with the number of iterations for the randomized grid search. The GNB model has no hyperparameters, so instead it simply selects the best feature selector, while KNN has a three other hyperparameters it searches over. That being said, given that this is a consistent pattern for multiple number of features, it might be related to the model rather than the randomized search.

Both Random Forest and KNN struggle to reach above 60% cross validation balance accuracy as the number of features increase, while the SVM model reach upwards of 83% at using 50 features and 72% at seven features. The GNB does

markedly improve its cross validation balanced accuracy to 76% at 50 features before plateauing. All models, however, fail to reach above 50% test balanced accuracy score baring a couple of outliers reaching at most 60%.

A similar pattern can be seen on mHDS trained data tested on mTCGA, which can be seen in figure 12. The main difference being that the models general fit better to the training data distribution, however, this is not surprising given the much smaller sample size.

4.3 miRNA Stage 2 + 3, Experiment (A)

A similar pattern can be seen on the miHDS trained models tested on miTCGA in figure 12. The distinction being that the models have better validation with fewer features than in figure 10, yet the poor generalization still persists. Note that the validation ROCs markedly become worse as more features are added, especially from 50 to 100 to 150 features. It is possible that the 25 feature model found a really good combination in the randomized grid search compared to that of 50, 100, and 150, or it could be equally likely that the additional features are noise not improving the model. The difference in training sample size is not significant, the miTCGA dataset consisting of 273 samples, hence that should not be a driving factor in the different cross validation ROCs compared to that of the mRNA study.

In summary, the models generalize poorly from TCGA to HDS and HDS to TCGA for all combinations of datasets, or in some cases generalize to the reverse relapse signal. A more detailed look at the difference in data distribution can be seen in section 4.4.

4.4 Difference Between Data Sources

The failure to generalize from TCGA to HDS or HDS to TCGA is interesting to note. One possible explanation is that the underlying distribution of the data or the labels are different between the two data sources. Consider firstly figure 13. The figure shows the Kaplan-Meier survival plot [28] for each data source for both overall survival (OS) and relapse. There is a definite discrepancy between the two data sources for both survival and relapse. Consider firstly relapse, as that is the response parameter for the analysis. There is significantly better relapse survival for stage 2 for the HDS dataset compared to TCGA and the relapse survival for stage 3 is worse for HDS compared to TCGA. The relapse survival for all stages are not that relevant given that the HDS source does not have any stage 4 patients, hence TCGA stage 4 survival will always be worse. Next consider the overall survival. Here HDS patients have a higher overall survival for both stage 2 and 3 compared to that of TCGA.

If there is any possible miss labelling of TNM stages such that samples that should be stage 1 or 4 are classified as stage 2 or 3, respectively, this should change the underlying signal given the assumption that cancer acts differently for different stages and that is detectable with a set of biomarkers. However, it seems unlikely that there is a mixup between stage 3 and 4, given the clear

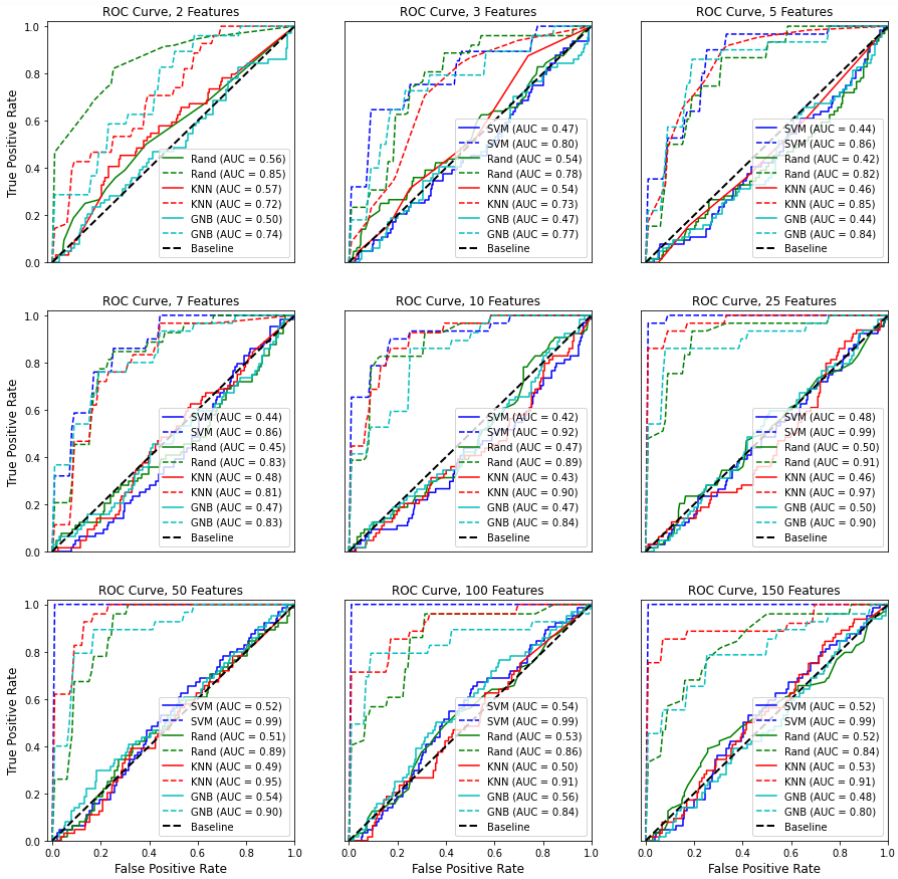

Figure 12: The figure shows the mean validation ROC on the miHDS dataset and the test ROC curves on the miTCGA dataset (experiment (A)). Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the miTCGA dataset. The plots show a clear failure to generalize from learning on HDS to predicting on TCGA. Furthermore, some of the test ROC curves are below the line of chance, indicating that whatever signal was found was found to be reversed in the TCGA dataset.

cut definition of stage 4 by metastasis. Another possibility is that the patients in the TCGA study happened to have poorer survival conditions post surgery, however, this is slightly confusing given that TCGA patients are on average 4.5 years younger than HDS patients, and increased age should have a correlation with other causes for death beyond the cancer itself. The relapse survival could be impacted by the possible post surgery treatment patients were given, how-

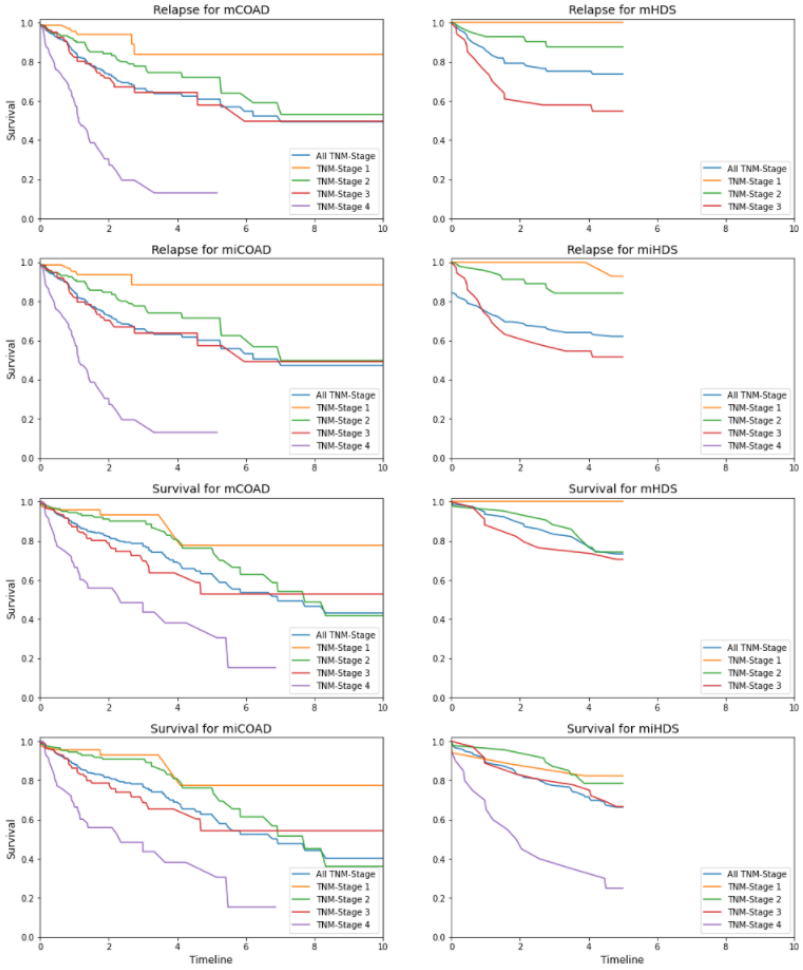

Figure 13: The figure shows the Kaplan-Meier [28] plots for both overall survival and relapse free interval for each data source. The subplot structure is setup such that it compares TCGA with HDS from left to right and data sets and relapse/survival downwards. The HDS study was limited to five years, while TCGA was limited to twelve, hence the difference in the charts. Note that only ten years are plotted for readability. TCGA has better relapse survival than HDS for stage 3, however, worse for both stage 2 and stage 1. The difference in all stages comes from the inclusion of stage 4 in the TCGA data which pulls the relapse survival down. On the other hand the overall survival for HDS is significantly better than TCGA for stage 2 and 3. It should be noted that the HDS dataset has very few samples and most patients die of old age rather than of the cancer, hence the difference in overall survival.

ever, that would imply whatever treatment the patients in the HDS dataset got compared to that of TCGA, the outcome was worse in terms of relapse and better in terms of survival. The standard procedure is to give chemotherapy to stage 3 patients for those part of the HDS cohort, however, that should in theory improve the relapse survival, not decrease it, compared to TCGA. That being said, the exact reasoning for this discrepancy is beyond the scope of this thesis, however, the difference could explain some of the poor generalization shown in section 4.2 and 4.3. Lastly, it is entirely possible that the true distribution of survival lies somewhere in-between the two cases shown above and that the small sample size leads to a high variance from the true data distribution, or one would converge to the other with increases sample size.

	miRNA		mRNA	
# Selected Genes (k)	Intersection	Union	Intersection	Union
2	0	32	0	27
3	1	40	0	41
5	3	63	0	77
7	4	87	0	112
10	6	114	0	157
25	26	241	3	352
50	77	384	9	637
100	189	587	26	1153
150	330	680	58	1654
250	572	755	158	2542

Table 3: The table shows the intersection and union of genes over all possible feature selector of size k for both data sources for a given data type (mRNA and miRNA). Formally the intersection is $\text{inter}_k(\text{mRNA}) = \left|\{\cup_{i=1}^{F}\text{Genes}_i^k(\text{mTCGA})\} \cap \{\cup_{i=1}^{F}\text{Genes}_i^k(\text{mHDS})\}\right|$ where $\text{Genes}_i^k(\text{mTCGA})$ is the specific selected gene names for gene selection method i that selects k number of genes. The union columns are defined similarly with a $\cap$ instead of $\cup$ between the two sources.

Another consideration is if there are features that are found in feature selection on both data sources. Table 3 shows the number of genes in the intersection and union of gene pool from the two data sources over all possible feature selector methods. Formally the intersection is $\text{inter}_k(\text{mRNA}) = \left|\{\cup_{i=1}^{F}\text{Genes}_i^k(\text{mTCGA})\} \cap \{\cup_{i=1}^{F}\text{Genes}_i^k(\text{mHDS})\}\right|$ where $\text{Genes}_i^k(\text{mTCGA})$ is the specific selected gene names for gene selection method i that selects k number of genes. The total number of different selection methods is detailed in section 3. This table does not account for how relevant a given feature is, only that at least one of the given models deems the feature important. The consequence of such a lax constraint is that features that have a considerable importance for both data sources are valued equally as features that are just tangentially relevant for a single feature selector. However, this the lax criteria is on purpose to see if there exists any overlap at all, no matter how insignificant.

	mTCGA			mHDS			miTCGA			miHDS		
	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other
0	TREML2	0.857	0.000	RAD17	0.571	0.000	hsa-miR-146a-3p	0.714	0.000	hsa-miR-181c-3p	0.357	0.000
1	FAM24B	0.500	0.000	CTAGE4	0.357	0.000	hsa-miR-6868-3p	0.429	0.000	hsa-miR-656-3p	0.357	0.214
2	TRIP10	0.214	0.000	SLC14A1	0.286	0.000	hsa-miR-656-3p	0.214	0.357	hsa-let-7g-3p	0.286	0.000
3	ZNF692	0.143	0.000	TPST2	0.214	0.000	hsa-miR-454-3p	0.214	0.000	hsa-miR-377-5p	0.286	0.000

	mTCGA			mHDS			miTCGA			miHDS		
	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other
0	TREML2	0.857	0.000	RAD17	0.571	0.000	hsa-miR-146a-3p	0.714	0.000	hsa-let-7g-3p	0.643	0.000
1	FAM24B	0.571	0.000	SHQ1	0.357	0.000	hsa-miR-6868-3p	0.429	0.000	hsa-miR-30d-5p	0.500	0.071
2	CATSPER3	0.214	0.000	CTAGE4	0.357	0.000	hsa-miR-504-5p	0.286	0.000	hsa-miR-656-3p	0.500	0.286
3	CCRL2	0.214	0.000	SLC14A1	0.286	0.000	hsa-miR-548f-3p	0.286	0.000	hsa-miR-181c-3p	0.357	0.000
4	TRIP10	0.214	0.000	TPST2	0.286	0.000	hsa-miR-656-3p	0.286	0.500	hsa-miR-377-5p	0.286	0.000
5	HOXB8	0.214	0.000	ZNF664	0.214	0.000	hsa-miR-337-5p	0.214	0.000	hsa-miR-3679-5p	0.286	0.000

	mTCGA			mHDS			miTCGA			miHDS		
	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other	Name	Presence	Presence Other
0	TREML2	0.857	0.000	RAD17	0.571	0.000	hsa-miR-146a-3p	0.714	0.000	hsa-let-7g-3p	0.714	0.000
1	FAM24B	0.571	0.000	SHQ1	0.429	0.000	hsa-miR-504-5p	0.500	0.000	hsa-miR-656-3p	0.643	0.429
2	CATSPER3	0.357	0.000	CTAGE4	0.357	0.000	hsa-miR-6868-3p	0.429	0.000	hsa-miR-30d-5p	0.571	0.143
3	NOX1	0.214	0.000	INPP5B	0.286	0.000	hsa-miR-656-3p	0.429	0.643	hsa-miR-576-5p	0.500	0.000
4	TRIP10	0.214	0.000	TPST2	0.286	0.000	hsa-miR-548f-3p	0.357	0.000	hsa-miR-376b-3p	0.429	0.143
5	RNF215	0.214	0.000	SLC14A1	0.286	0.000	hsa-miR-548l	0.286	0.000	hsa-miR-5010-3p	0.357	0.000
6	CCRL2	0.214	0.000	IGFBP3	0.214	0.000	hsa-miR-454-3p	0.286	0.000	hsa-miR-181c-3p	0.357	0.000
7	MPZ	0.214	0.000	ZNF664	0.214	0.000	hsa-miR-337-5p	0.214	0.000	hsa-miR-676-3p	0.286	0.000

Figure 14: The figure shows a table of the presence of a given gene in feature selectors for a k equal to 3, 5, and 7. The presence column indicate the proportion of feature selectors that selected said feature and the presence other is the proportion of feature selectors that selected said feature in the other data source. Specifically, take the top table, TREML2 has a presence of 0.857 in mTCGA and 0 in mHDS. Similarily RAD17 has a 0.571 presence in mHDS and 0 presence in mTCGA. For each dataset the genes are sorted by their presence. Only the $k + 1$ top features are shown. The tables show that the presence of mRNA genes is not replicated in the other data source, while for miRNA genes there is some moderate presence in the other data source.

The table clearly shows that there is virtually zero overlap for mRNA and some overlap for miRNA. The overlap for miRNA quite likely comes from the fact that there are ten times fewer features in the miRNA dataset compared to the mRNA datasets. The lack of overlap helps explain why the models generalize poorly, since the genes found in one data source do not hold predictive power in the other source.

Instead of looking at just the size of the intersection of genes, consider figure 14 that shows a list of the top $k+1$ number of genes ranking by what percentage of feature selectors the given feature is selected in. The figure shows feature selectors of size three, five, and seven.

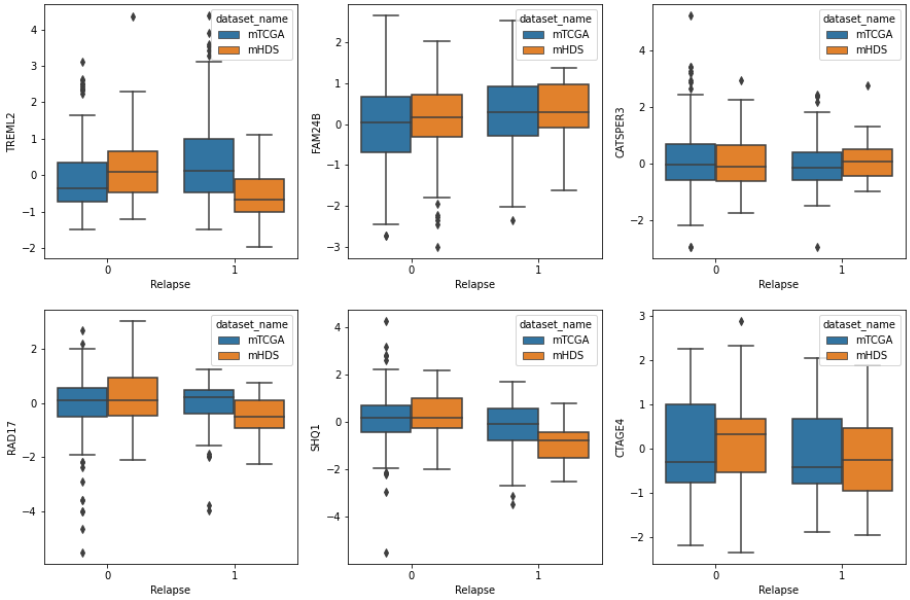

Figure 15: The figure shows the boxplot distribution for relapse and no relapse for each mTCGA and mHDS for a select number of features. The blue plots are for mTCGA and orange for mHDS. The box represents the middle 50% of the data and the line through it the median. The whiskers represent the outlier boundary and the outliers are indicated as data points. The first row of figures show the distribution for the three genes that have the highest presence in the mTCGA feature selectors for $3 < k < 25$. Similarity the second row are genes that are prominent in the mHDS feature selectors. The range of $3 < k < 25$ was selected because that is roughly the range of desirable number of features for a model. A higher k would also median that genes that have a very weak signal that many models could pick up would be part of the list.

Another possible approach to identifying the poor generalization is to look at the data distribution of each class for both data sources for features that are selected to perform well, see figure 15. This can be shown as a boxplot, i.e. a box showing the 25-75 percentile and the whiskers showing the boundary for outliers, < 5 percentile or > 95 percentile, side by side. Note that the class data distribution is shown, instead of the feature distribution, since what is important is how the data is distributed for each class comparatively instead of the total data distribution, which is, given the standardization step, mean 0 and standard deviation 1. If a feature has the same predictive signal across both data

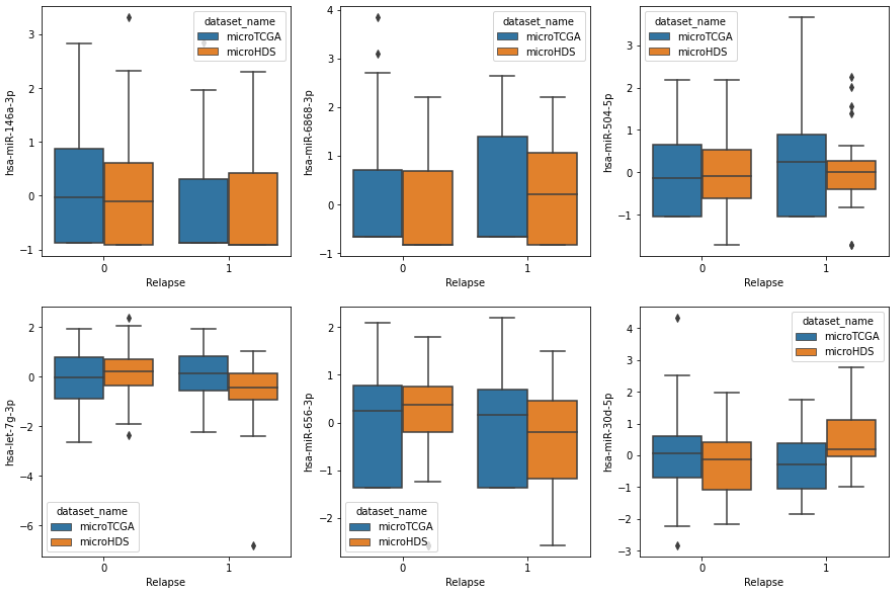

Figure 16: The figure shows the boxplot distribution for positive and negative relapse for each miTCGA and miHDS for a select number of features. See figure 15 for a more detailed description of the figure type.

sources it should have the same signal distribution. I.e. if a gene is positively correlated with relapse, then it should still be positively correlated with relapse in the other data source. If the other data source has a negative correlation, that would explain why predictions can be flipped, as a negative correlation would simply mean it is correlated with no relapse instead of relapse. If the relapse distribution is not markedly different from the no relapse distribution, i.e. the 50 quantile is quite wide and intersects the domain of the relapse 50 quantile, then the feature is simply inconclusive and can lead to false positives and false negatives, rather than simply almost always predicting the wrong class.

Consider figure 15. The TREML2 gene, which is among the highest presence in mTCGA feature selectors, show exactly how the signal of the two data sources can be different. TREML2 is negatively correlated with relapse in mHDS while it is positively correlated in mTCGA. This means that models that rely on TREML2 will not generalize well on mHDS samples. However, CASPER3 have relatively the same signal for relapse for both data sources, albeit the total spread of relapse samples are smaller than for non relapse samples. There is a small difference between the two data sources given that CASPER3 is slightly positively correlated with relapse in mHDS, but it is nowhere near as stark of a difference as for TREML2. Another problem with CASPER2 is that the class distribution within a single data source is not that different. The mean and spread of mTCGA is relatively similar, even if the outliers are slightly more

spread out for non relapse samples. Genes that follow this general distribution would be a poor univariate estimator compared to that of TREML2. This is important because not all genes that a feature selector selects are part of a feature selector set. For instance if SVM_l1 selects TREML2 and another gene, that combined are have better predictive power than alone, the two genes need not be a part of the feature set with k number of features. On the second row both RAD17 and SHQ1 show a similar pattern to TREML2 in that the signal is vastly different between the two data sources. Note that the second row are genes that the feature selectors deem relevant on the mHDS dataset, hence the negative correlation with relapse is what the feature selectors deemed relevant.

Figure 16 shows the boxplot for six selected miRNAs. Note that, unlike for mRNA, the miRNA "hsa-miR-656-3p" has a high presence in both data sources, but the other five have no presence on the opposite data source feature selection pool. It is interesting that this gene is important for both selectors given the class signal appears identical for relapse and non relapse samples for miTCGA. Another thing to note is that the relapse distributions for "hsa-miR-146a-3p" have a mean at the lowest point of the distribution. A similar pattern can be seen for "hsa-miR-686-3p", but for non relapse samples instead of relapse samples.

4.5 mRNA + miRNA for Stage 2 + 3, Experiment (D)

Next consider the experiment of combining both mRNA and miRNA into a single feature space. This does slightly increase the dimensionality relative to just mRNA, however, the increase is marginal at best, about a 6% increase at most. This allows the feature selector to find combinations of mRNAs and miRNAs that help explain the relapse distribution. This experiment was done only on TNM-stage 2 and 3 sample.

Figure 17 show the results of training on TCGA data and testing on HDS. The figure shows the same pattern of models increasing cross validation ROC as the number of features increases, however, it does show that the generalization actually improves with more features, specifically becoming markedly better at 25 features. This indicates that there is at least some congruent relapse signal between the two data sources, even if it does mean both mRNAs and miRNAs have to be included to find such a signal. Additionally, the KNN continues to lag behind the other models, presumably struggling with noisy additional features as the number of selected features increases. Even if the test ROC curves do improve with more features, the balanced accuracy measure remains mostly unchanged, hovering around the baseline of 50% indicating that the default probability decision boundary could be tuned such that the model performs better. On the other hand figure 18 show that the strong cross validated signal that is found in HDS is not replicated when testing on TCGA, and the balanced accuracy measure is no better either.

That being said, figure 19 show that the features that are selected is a mix of mRNAs and miRNAs, note the different naming convention for miRNAs including a "miR" part and/or "-3p" or "-5p" suffix. However, the figure also

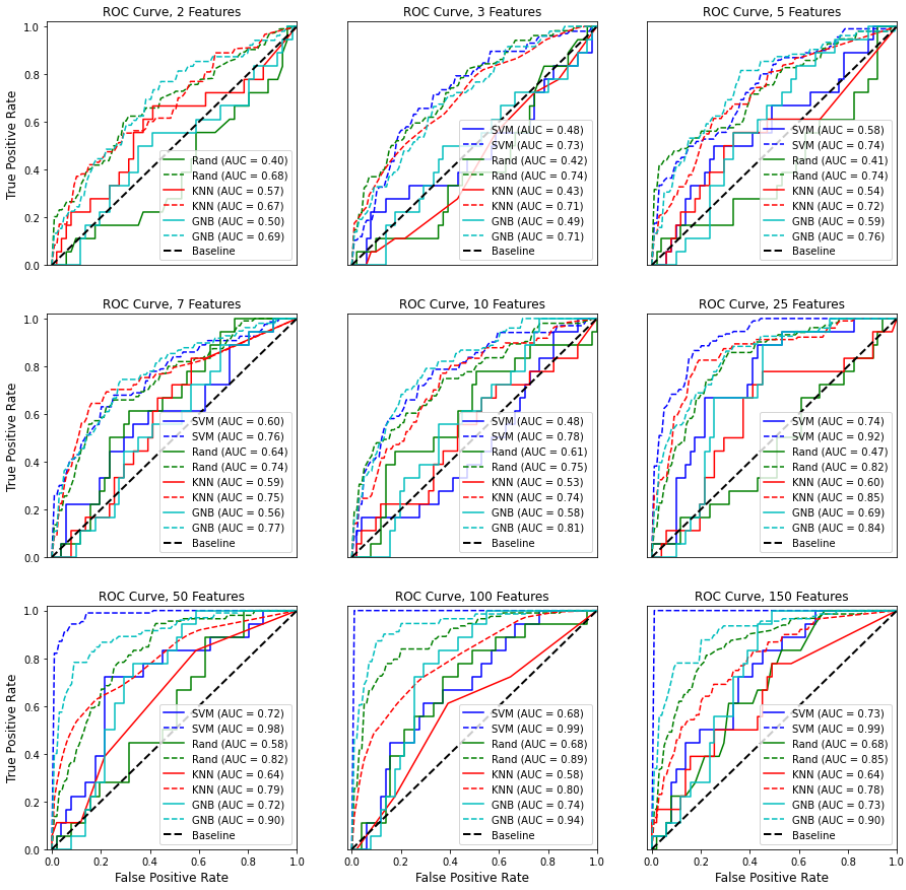

Figure 17: The figure shows the mean validation ROC on the combined TCGA dataset and the test ROC curves on the combined HDS dataset. Combined means that the feature space from the mRNA and miRNA datasets have been concatenated prior to feature selection. Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the HDS dataset. The plots show a clear failure to generalize from learning on HDS to predicting on TCGA.

shows that the presence in the other data source remains non existent, hence partially explaining why the HDS trained models fail to generalize on TCGA data.

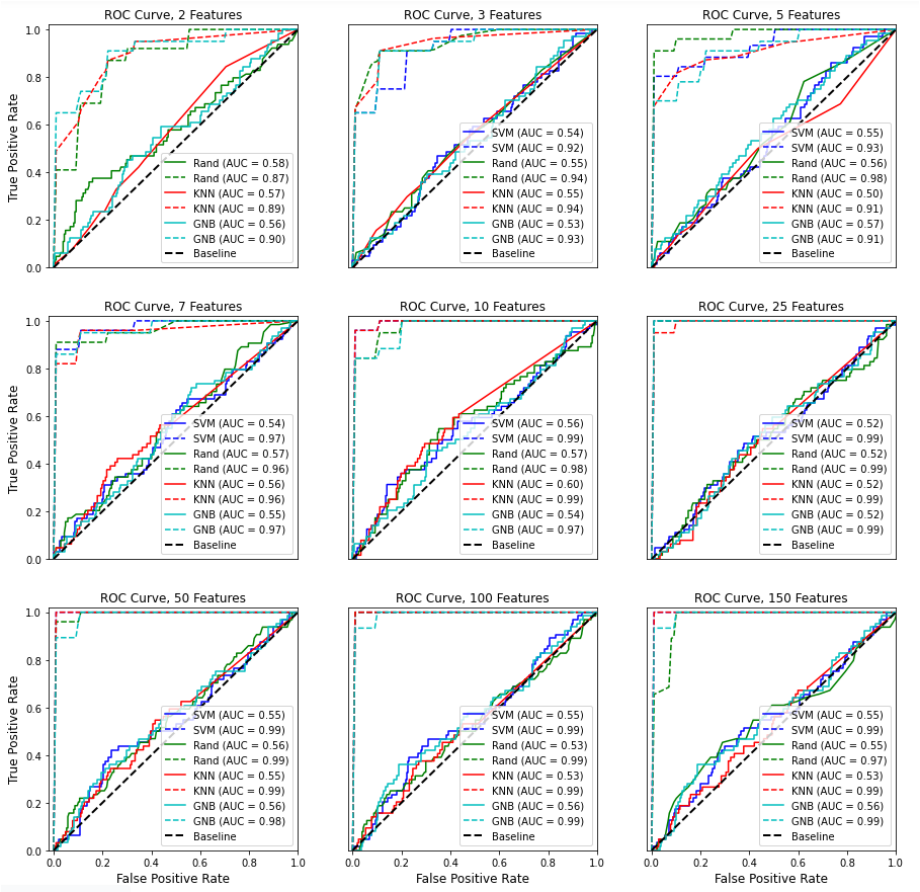

Figure 18: The figure shows the mean validation ROC on the combined HDS dataset and the test ROC curves on the combined TCGA dataset (experiment (D)). Combined means that the feature space from the mRNA and miRNA datasets have been concatenated prior to feature selection. Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the TCGA dataset. The plots show a clear failure to generalize from learning on HDS to predicting on TCGA.

4.6 mRNA or miRNA Combined with Clinical Data, Experiment (E)

Experiment E combines a dataset with the clinical parameters after performing feature selection, and a model is trained on the new feature space. The best performing dataset in terms of generalization is shown in figure 20. The figure shows that the Gaussian Naive Bayes classifier reaches a peak performance near

	HDS			TCGA		
	Name	Presence	Presence Other	Name	Presence	Presence Other
0	SHQ1	0.357	0.000	TREML2	0.714	0.000
1	SLC14A1	0.357	0.000	hsa-miR-146a-3p	0.357	0.000
2	hsa-miR-4762-5p	0.286	0.000	FAM24B	0.286	0.000
3	hsa-miR-887-5p	0.214	0.000	ZNF440	0.143	0.000

	HDS			TCGA		
	Name	Presence	Presence Other	Name	Presence	Presence Other
0	RAD17	0.357	0.000	TREML2	0.929	0.000
1	SHQ1	0.357	0.000	hsa-miR-146a-3p	0.500	0.000
2	SLC14A1	0.357	0.000	FAM24B	0.429	0.000
3	hsa-miR-4762-5p	0.286	0.000	hsa-miR-548l	0.286	0.000
4	hsa-miR-676-3p	0.214	0.000	CCRL2	0.214	0.000
5	hsa-miR-4684-3p	0.214	0.000	ZNF597	0.143	0.000

	HDS			TCGA		
	Name	Presence	Presence Other	Name	Presence	Presence Other
0	RAD17	0.571	0.000	TREML2	0.929	0.000
1	hsa-miR-676-3p	0.429	0.000	hsa-miR-146a-3p	0.500	0.000
2	SLC14A1	0.357	0.000	hsa-miR-548l	0.429	0.000
3	SHQ1	0.357	0.000	FAM24B	0.429	0.000
4	RAB3D	0.286	0.000	ZNF597	0.214	0.000
5	hsa-miR-4762-5p	0.286	0.000	CCRL2	0.214	0.000
6	hsa-miR-4684-3p	0.286	0.000	RHOBTB1	0.214	0.000
7	DNASE1L1	0.214	0.000	BDNF	0.214	0.000

Figure 19: The figure shows a table of the presence of a given gene in feature selectors for a k equal to 3, 5, and 7 when the mRNA and miRNA data is concatenated prior to feature selection. The presence column indicate the proportion of feature selectors that selected said feature and the presence other is the proportion of feature selectors that selected said feature in the other data source. For each data source the genes are sorted by their presence. Only the $k + 1$ top features are shown.

7 features, reaching a 70% imbalanced accuracy score, markedly better than any of the other trained models, however, even a 2 feature model still perform almost equally as well. This indicates that it is possible adding clinical parameters to the data would improve the model, however, this pattern does not repeat for any other model, nor for any other dataset, thus making the inclusion dubious. This is further supported by the clear difference in age, which has been shown to allow models to generalize better from TCGA to HDS.

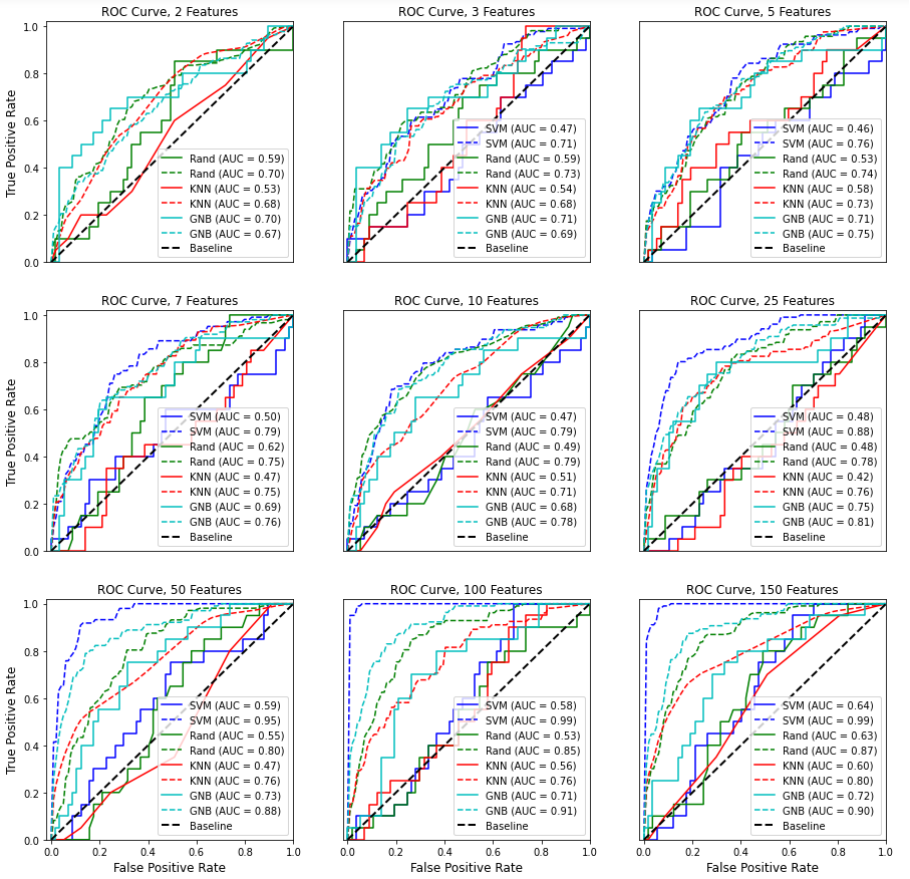

Figure 20: The figure shows the mean validation ROC on the combination of mTCGA data and clinical data after feature selection has been performed on mTCGA data (experiment (E)). Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the mHDS dataset.

4.7 Transcriptome Data All Stages, Experiment (B)

Lastly, consider the case for mRNA and miRNA data for all stages. Despite increasing the sample size when including stage 1 and 4 samples, the models remain mostly unchanged performing near the line of chance. One of the four cases, trained on TCGA and tested on HDS, is shown in figure 21. Of the four datasets, this is the highest performing generalization ROC, however, the models have poor balanced accuracy score near the baseline of 50%. The other cases are not shown, given the case shown in figure 21 is the best case scenario for experiment (B).

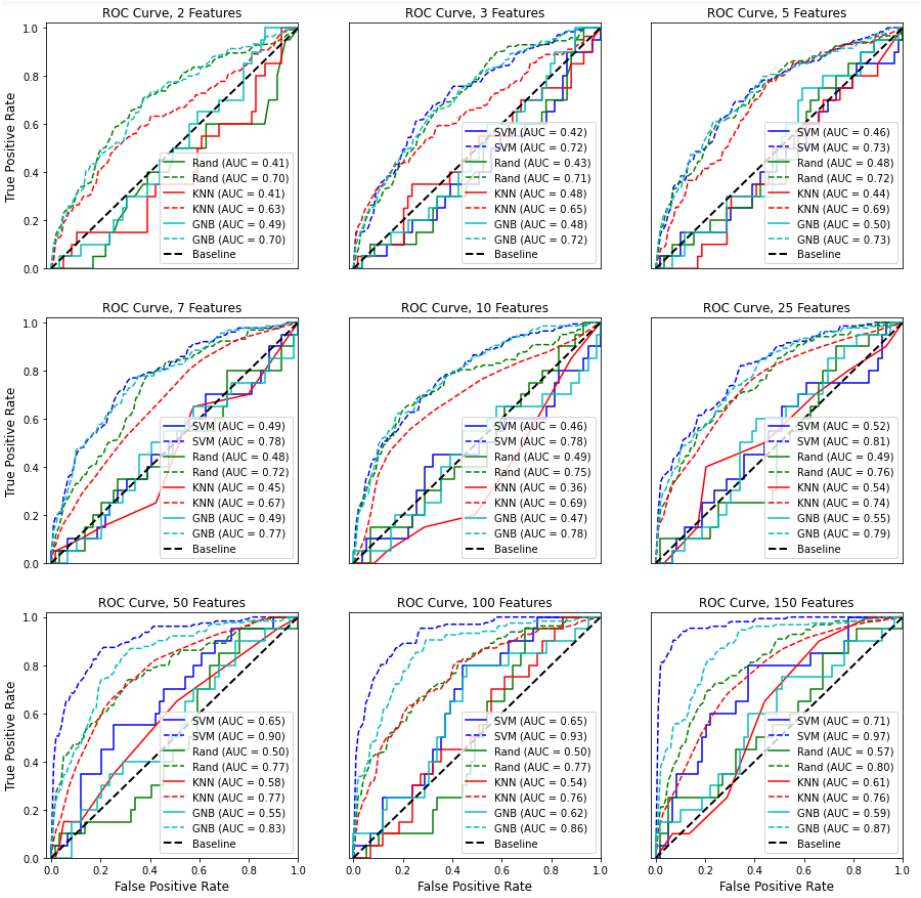

Figure 21: The figure shows the mean validation ROC on the mTCGAA dataset and the test ROC curves on the mHDS dataset for all TNM stages (experiment (B)). Each subplot contains the ROC curves for models using k number of selected features from left to right in ascending order. The dotted lines are the mean validation ROC curves and the solid lines are the test ROC curves on the mHDS dataset.

4.8 Other Attempted Methods and Experiments

There are other factors than mRNA and miRNA that might be relevant to recurrence, one of which being the proportion of different cell types present in a tumor sample. The following researches have compiled a number of methods to calculate the infiltration estimate, i.e. proportion of cells within a tumor, for all TCGA datasets [31]. There is also a possibility of predicting the infiltration estimate for new datasets as well as part of their website TIMER2.0. The infiltration estimates are based on a number of different methods and have potential

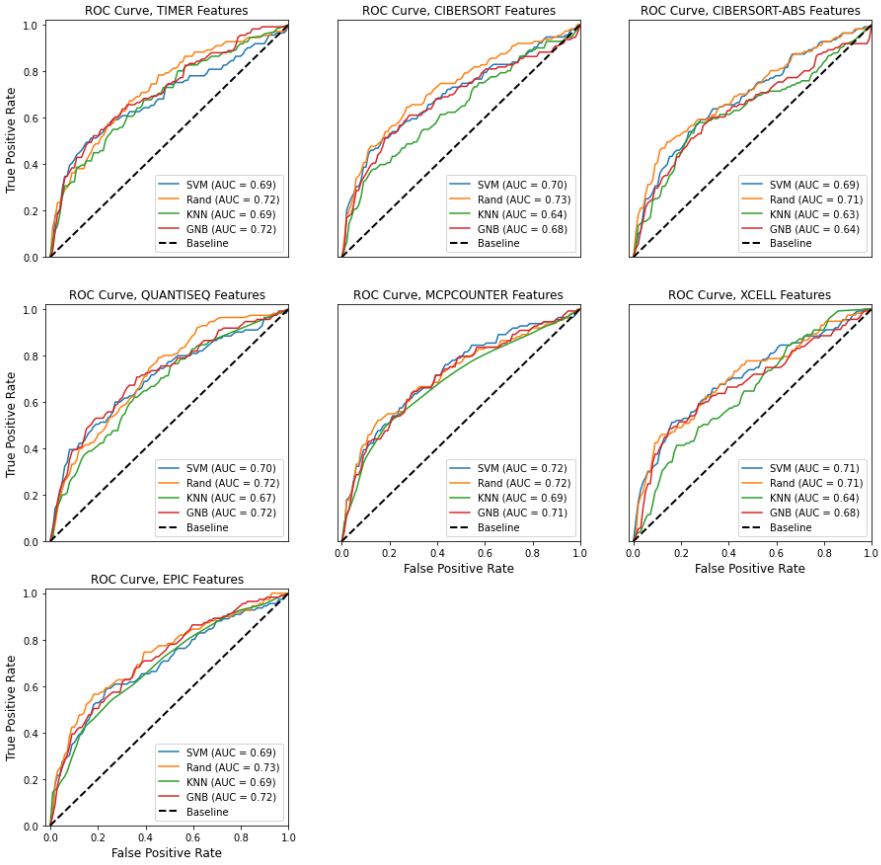

Figure 22: The figure shows the mean validation ROC for models trained on both clinical data and each of the seven different infiltration estimate models individually for each subplot.

to being relevant as an additional datapoint to clinical data, however, the few experiments that was conducted on the TCGA data did not yield and significant results and models struggled with the class imbalance, hence the approach was scrapped. Figure 22 shows the preliminary results of those experiments, showcasing the models not finding any relevant signal. Note that there is no test set for this analysis given that the infiltration estimate for HDS was not available.

Another approach for feature selection was to use a Genetic Algorithm to select subsets of features that was used to train SVM models with l1 weight decay as a means to perform SVM l1 feature selection on subselections of the total feature space. This appraoch was also scrapped due to problems with the initialization of the subsets and having the algorithm properly explore a wide subset of features. The models instead just hovered around the same feature

space or simply expanded to include more features, which was not conducive. The approach was based on a paper using Grasshopper Optimization as a meta-heuristic learning approach to feature selection [47].

5 Concluding Remarks and Future Work

The results for the trained models on one data source shows a failure to generalize to the other. In the few cases where there are some reasonable test ROC curves, the predictions are still off, meaning that the probability decision boundary needs to be tweaked. Combining both mRNA and miRNA before performing feature selection yields better generalization for the downstream models, even if the presence across data sources for said selected features remain non existent. Combining transcriptome and clinical data show some promise for one model when trained on mTCGA and tested on mHDS, however, all other datasets fail. The best models were trained on clinical data in terms of generalization performance, however, that can be questioned given the reliance on age. Three problems have been identified in this study something something The problems can be summarized in three key points 1) opposite or different signals between the two data sources, 2) single step feature selection, and 3) class imbalance.

Problem 1) is shown in figure 15 and discussed at length in section 4.4. This is related to how the data was collected, how it was preprocessed before the analysis, or possibly that the features found are not important. Given the relative uncertainty here it might be beneficial to perform the same analysis on a different data source to determine if the same pattern repeats. The paper [43] performed non-negative matrix factorization on mRNA data from two Gene Expression Omnibus (GEO) [12] datasets that were uploaded to the National Center for Biotechnology Information (NCBI) website related to the following two papers [27] and [26]. Alternatively an analysis could be performed on some library adjusted raw count values to determine if these genes have a different relapse signal before the normalization step.

Problem 2) is the single step feature selection technique that has a weakness to overfitted wrapper models. With a sufficient number of features from an overfitted wrapper model, a model trained on said features could replicate the existing overfitted model. There are some indications of this for the SVM model, however, it is not a consistent problem. Additionally, the filter methods also struggle to find statistically significant features based on the adjusted p-values, hence why the k most important features are selected instead.

One remedy could be to perform a nested feature selection approach. Use the filter based method to filter down to a large number of features, then take the union over those feature spaces, and perform wrapper methods on that reduced feature space. The advantage of this approach is that the wrapper methods have less features to heavily overfit on and the subset selection can be based on forward/backward selection methods instead of a single trained model.

In section 4.8 a Genetic Algorithm [40] approach that was abandoned due to time constraints was mentioned. This type of approach to explore the full

feature space via smaller subspaces of features that are selected based on a metaheuristic could prove powerful as a feature selector. This would be similar to how Random Forest reduces bias by randomizing features for each of its estimators, albeit the difference being that a specific estimator is selected over using all of them in an ensemble. However, the latter could be possible. Consider a ranking system of each feature based on its importance in a model and the relative performance of its associated wrapper method, then use the combined score over the whole feature space to determine good features. One problem with such an approach is that good features would be drowned out by the many bad features unless proper care is taken to account for model performance based on the given features.

Sharifai and Zainol [47] used a grasshopper optimization algorithm to select subsets of features that was used to train individual SVM models to select features. Many other metaheuristic approaches could be applied as well, many of which are implemented in the mealpy python package for use [50].

Regardless of how feature selection is performed, it is important to note that for mRNA data there is a huge feature space compared to the sample space, hence models for feature selection needs to take this into account. Even a set of filter methods into wrapper methods on the union still runs into the high dimensionality problem for the filter methods. Approaches like a genetic algorithm, or other metaheuristics, still need to explore a relatively large feature space and models need to be properly penalized for using unimportant features.

Lastly, from problem 3), the class imbalance problem could be partially alleviated by performing random over-sampling when performing the randomized grid search to determine optimal hyperparameters. It should be noted that with sufficiently small feature spaces, it is also possible to perform SMOTE or ADASYN sampling, however, care should be taken to the bias in the oversampling, see [30]. The same could be said for random over sampling, which simply duplicates data samples.

For future work I recommend using additional datasets from the GEO database to determine if the difference in gene expression exist for those datasets as well. If that fails suggest using gene mutation or DNA methylation data which is available for both TCGA and HDS as additional sources of information. The methods fail on the mRNA and miRNA data because there is a difference in the data distribution, hence trying some other sources of data might yield better results.

References

[1] Helsedirektoratet (2017). Nasjonalt handlingsprogram med retningslinjer for diagnostikk behandling og oppfølging av kreft i tykktarm og endetarm [nettdokument]. Oslo: Helsedirektoratet (sist faglig oppdatert 04. mai 2021, lest 31. mai 2021). Tilgjengelig fra `https://www.helsedirektoratet.no/retningslinjer/kreft-i-tykktarm-og-endetarm-handlingsprogram`.

[2] Pankaj Ahluwalia, Ravindra Kolhe, and Gagandeep K Gahlay. The clinical relevance of gene expression based prognostic signatures in colorectal cancer. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, page 188513, 2021.

[3] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander D Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential cell biology*. Garland Science, 2015.

[4] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? pages 217–235, 1999.

[5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. pages 144–152, 1992.

[6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] Chiara Cremolini, M Di Bartolomeo, A Amatu, Carlotta Antoniotti, R Moretto, R Berenato, F Perrone, E Tamborini, G Aprile, Sara Lonardi, et al. Braf codons 594 and 596 mutations identify a new molecular subtype of metastatic colorectal cancer at favorable prognosis. *Annals of oncology*, 26(10):2092–2097, 2015.

[10] Ana Custodio and Jaime Feliu. Prognostic and predictive biomarkers for epidermal growth factor receptor-targeted therapy in colorectal cancer: beyond kras mutations. *Critical reviews in oncology/hematology*, 85(1):45–81, 2013.

[11] Cameron Davidson-Pilon, Jonas Kalderstam, Noah Jacobson, Sean Reed, Ben Kuhn, Paul Zivich, Mike Williamson, AbdealiJK, Deepyaman Datta, Andrew Fiore-Gartland, Alex Parij, Daniel WIlson, Gabriel, Luis Moneda, Arturo Moncada-Torres, Kyle Stark, Harsh Gadgil, Jona, JoseLlanes, Karthikeyan Singaravelan, Lilian Besson, Miguel Sancho Peña, Steven Anton, Andreas Klintberg, GrowthJeff, Javad Noorbakhsh, Matthew Begun,

Ravin Kumar, Sean Hussey, and Skipper Seabold. Camdavidsonpilon/lifelines: 0.26.0, May 2021.

[12] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

[13] Stephen B. Edge and American Joint Committee on Cancer, editors. *AJCC cancer staging manual*. Springer, New York, 8th ed edition, 2017.

[14] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[15] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.

[16] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[17] Eric R Fearon. Molecular genetics of colorectal cancer. *Annual Review of Pathology: Mechanisms of Disease*, 6:479–507, 2011.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[19] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

[20] Justin Guinney, Rodrigo Dienstmann, Xin Wang, Aurélien De Reyniès, Andreas Schlicker, Charlotte Soneson, Laetitia Marisa, Paul Roepman, Gift Nyamundanda, Paolo Angelino, Brian M. Bot, Jeffrey S. Morris, Iris M. Simon, Sarah Gerster, Evelyn Fessler, Felipe De Sousa .E Melo, Edoardo Missiaglia, Hena Ramay, David Barras, Krisztian Homicsko, Dipen Maru, Ganiraju C. Manyam, Bradley Broom, Valerie Boige, Beatriz Perez-Villamil, Ted Laderas, Ramon Salazar, Joe W. Gray, Douglas Hanahan, Josep Tabernero, Rene Bernards, Stephen H. Friend, Pierre Laurent-Puig, Jan Paul Medema, Anguraj Sadanandam, Lodewyk Wessels, Mauro Delorenzi, Scott Kopetz, Louis Vermeulen, and Sabine Tejpar. The consensus molecular subtypes of colorectal cancer. *Nature Medicine*, 21:1350–1356, 11 2015.

[21] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *Cell*, 144(5):646–674, 2011.

[22] Wolfgang Huber, Anja von Heydebreck, Holger Sueltmann, Annemarie Poustka, and Martin Vingron. Parameter estimation for the calibration and variance stabilization of microarray data. *Statistical Applications in Genetics and Molecular Biology*, 2, 12 2005.

[23] Havjin Jacob, Luka Stanisavljevic, Kristian Eeg Storli, Kjersti E Hestetun, Olav Dahl, and Mette P Myklebust. Identification of a sixteen-microrna signature as prognostic biomarker for stage ii and iii colon cancer. *Oncotarget*, 8(50):87837, 2017.

[24] Havjin Jacob, Luka Stanisavljevic, Kristian Eeg Storli, Kjersti E. Hestetun, Olav Dahl, and Mette P. Myklebust. A four-microrna classifier as a novel prognostic marker for tumor recurrence in stage ii colon cancer. *Scientific Reports*, 8, 12 2018.

[25] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[26] Robert N Jorissen, Peter Gibbs, Michael Christie, Saurabh Prakash, Lara Lipton, Jayesh Desai, David Kerr, Lauri A Aaltonen, Diego Arango, Mogens Kruhøffer, et al. Metastasis-associated gene expression changes predict poor outcomes in patients with dukes stage b and c colorectal cancer. *Clinical Cancer Research*, 15(24):7642–7651, 2009.

[27] Robert N Jorissen, Lara Lipton, Peter Gibbs, Matthew Chapman, Jayesh Desai, Ian T Jones, Timothy J Yeatman, Philip East, Ian PM Tomlinson, Hein W Verspaget, et al. Dna copy-number alterations underlie gene expression differences between microsatellite stable and unstable colorectal cancers. *Clinical Cancer Research*, 14(24):8061–8069, 2008.

[28] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[29] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, Jun 2004.

[30] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.*, 18(1):559–563, January 2017.

[31] Bo Li, Eric Severson, Jean-Christophe Pignon, Haoquan Zhao, Taiwen Li, Jesse Novak, Peng Jiang, Hui Shen, Jon C. Aster, Scott Rodig, Sabina Signoretti, Jun S. Liu, and X. Shirley Liu. Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome Biology*, 17(1), August 2016.

[32] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

[33] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[34] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), December 2014.

[35] N. Marschner, M. Frank, W. Vach, E. Ladda, A. Karcher, S. Winter, M. Jänicke, and T. Trarbach. Development and validation of a novel prognostic score to predict survival in patients with metastatic colorectal cancer: the metastatic colorectal cancer score (mccs). *Colorectal Disease*, 21:816–826, 7 2019.

[36] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[37] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. 2020.

[38] Shuji Ogino, Katsuhiko Nosho, Gregory J Kirkner, Kaori Shima, Natsumi Irahara, Shoko Kure, Andrew T Chan, Jeffrey A Engelman, Peter Kraft, Lewis C Cantley, et al. Pik3ca mutation is associated with poor prognosis among patients with curatively resected colon cancer. *Journal of clinical oncology*, 27(9):1477, 2009.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] Stjepan Picek and Marin Golub. Comparison of a crossover operator in binary-coded genetic algorithms. *WSEAS transactions on computers*, 9(9):1064–1073, 2010.

[41] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[42] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 11 2009.

[43] Anguraj Sadanandam, Costas A. Lyssiotis, Krisztian Homicsko, Eric A. Collisson, William J. Gibb, Stephan Wullschleger, Liliane C.Gonzalez Ostos, William A. Lannon, Carsten Grotzinger, Maguy Del Rio, Benoit Lhermitte, Adam B. Olshen, Bertram Wiedenmann, Lewis C. Cantley, Joe W. Gray, and Douglas Hanahan. A colorectal cancer classification system that associates cellular phenotype and responses to therapy. *Nature Medicine*, 19:619–625, 5 2013.

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[45] D. J. Sargent, S. Marsoni, S. N. Thibodeau, R. Labianca, S. R. Hamilton, V. Torri, G. Monges, C. Ribic, A. Grothey, and S. Gallinger. Confirmation of deficient mismatch repair (dmmr) as a predictive marker for lack of benefit from 5-fu based chemotherapy in stage ii and iii colon cancer (cc): A pooled molecular reanalysis of randomized chemotherapy trials. *Journal of Clinical Oncology*, 26(15_suppl):4008–4008, 2008. PMID: 27949263.

[46] Daniel J Sargent, Qian Shi, Greg Yothers, Sabine Tejpar, Monica M Bertagnolli, Stephen N Thibodeau, Thierry Andre, Roberto Labianca, Steven Gallinger, Stanley R Hamilton, et al. Prognostic impact of deficient mismatch repair (dmmr) in 7,803 stage ii/iii colon cancer (cc) patients (pts): A pooled individual pt data analysis of 17 adjuvant trials in the accent database., 2014.

[47] Garba Abdulrauf Sharifai and Zurinahni Zainol. Feature selection for high-dimensional and imbalanced biomedical data based on robust correlation based redundancy and binary grasshopper optimization algorithm. *Genes*, 11:1–26, 7 2020.

[48] Marko Robnik- Sikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff, 2003.

[49] Ole Johan Skrede, Sepp De Raedt, Andreas Kleppe, Tarjei S. Hveem, Knut Liestøl, John Maddison, Hanne A. Askautrud, Manohar Pradhan, John Arne Nesheim, Fritz Albregtsen, Inger Nina Farstad, Enric Domingo, David N. Church, Arild Nesbakken, Neil A. Shepherd, Ian Tomlinson, Rachel Kerr, Marco Novelli, David J. Kerr, and Håvard E. Danielsen. Deep learning for prediction of colorectal cancer outcome: a discovery and validation study. *The Lancet*, 395:350–360, 2 2020.

[50] Nguyen Van Thieu. A collection of the state-of-the-art meta-heuristics algorithms in python: Mealpy, 2020.

[51] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. pages 856–863, 2003.

[52] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. page 116, 2004.

[53] Yongli Zhang and Yuhong Yang. Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112, 2015.