Identification of Tumor Patients with Deep Learning Approaches Based on microRNAs Measured in Blood Samples

Harry Ritchie
Machine Learning Seminar
Chair for Clinical Bioinformatics
Saarland University, Saarbrucken, Germany

***Abstract*—Predicting cancer is no easy challenge [1]. Dealing with human lives requires models and algorithms, not only accurate, that also reduce the number of false negatives. The rise of Deep Learning has brought many powerful models to the table, reducing the risk of false negatives, however, more common in the field of medical imaging [2]. Recall or sensitivity is of paramount importance. From a lung cancer dataset of 3046 samples and 1183 MicroRNA features, we tested two Deep Neural Networks (DNN) named, DeepNet2 and DeepNet3. Comparing and contrasting the models, it's revealed the dataset is too small, and imbalanced (80% Non-lung cancer, 20% lung cancer). Support Vector Machines (SVM) championed the problem combined with over-sampling. Our SVM produced an AUC value of 0.98 and sensitivity of 0.94 evaluated with Leave-One-Out Cross Validation.**

I. Introduction

In 2019, there are 1,762,450 new cancer cases in the US alone [3]. Given the sheer number of cases in the US, many go undiagnosed. The introduction of Artificial Intelligence (AI) has allowed predictive technology to ease the early diagnosis of cancer [4]. One such new diagnostic measure is with the use of MicroRNAs. They are non-coding RNAs that have been implicated in a number of human diseases [5]. It has been shown MicroRNAs are highly stable in blood, thus requiring only blood samples for further analysis [6]. Utilising Machine Learning techniques can not only describe how diseases are reflective of the data, they can also predict whether a patient has the disease, or not. In recent years, the topic of "Deep Learning" has caught significant traction, and has proved itself as a powerful topic in Machine Learning [7]. The name Deep Learning owes itself to the many layers found in neural networks, where more than two "hidden" layers are present – some would argue greater than three [8]. In this report, we present a comparison of multiple approaches to find a model best suited to the problem.

Over-fitting is a common problem in machine learning. Models attempt to explain variances across all features, and given many features, predictive performance becomes too specific for the training data. A clear identifier of over-fitting can be visualised when plotting training error vs test error.

II. Material and Methods

A. Dataset

The original data, labelled $\mathbb{D}$, consists of 3046 microRNA samples with 1183 positive continuous features.

$$\mathbb{D} \in \mathbb{R}^{3046 \times 1183}$$

The response is binary, whether the patient has lung cancer (1), or not (0).

$$\mathbf{y} \in \{0, 1\}$$

We aim to find a function f, such that,

$$f(\mathbb{D}) = \mathbf{y}$$

The data was class imbalanced, where 80% of the samples belong to Non-LCa ($\mathbb{D}_-$) and 20% to LCa ($\mathbb{D}_+$) .

$$|\mathbb{D}_+| = 606$$
$$|\mathbb{D}_-| = 2440$$

B. Inspection

As a preliminary inspection, a report summary of the dataset provides any initial insight into the data. The data was then split into two groups, LCa and non-LCa, to describe each feature. The data itself is normalised, allowing each feature to be compared.
To establish any priors insight, for each feature, the average, standard deviation, 25, 50, 75% quantile, minimum and maximum values were calculated.
Testing for linear dependence, a correlation heatmap was produced; granting us a basis to work with.

C. Splitting

The data was split 75% training and 25% test set. Stratified splitting accounted for class balance to maintain equal samples for learning. The training set was over-sampled to increase proportion of LCa samples.

D. Preprocessing

Two normalisation procedures were used for specific reasons, standardisation and scaling of variables to the range $\{0, 1\}$.

Scaling to $\{0, 1\}$ is defined as:

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

Standardisation is defined as:

$$z = \frac{x - \mu}{\sigma}$$

Due to the large fluctuation of values, a log transform was applied on the data.
The data was standardised to ensure faster convergence for tree-models (Gradient Boosting, Random Forest, Bagging), however, was not necessary for Logistic Regression and Suport Vector Machines (SVM). The Neural Network's input variables were scaled to the range $\{0, 1\}$.

As part of the visualisation procedure, we applied PCA to project the standardised data to two dimensions. This is however under the assumption the data is linear, due to its linear combinations of the input variable $\mathbf{X}$.

An additional, non-linear procedure, t-Stochastic Neighbourhood Embedding (t-SNE), in an attempt to spread classes for visualisation the data, was also applied. T-SNE is a probabilistic approach displaying points that a closer with a certain probability compared to those further away. The setback of this technique requires high familiarity of the problem and understanding of t-SNE parameters.

E. Feature Selection

Feature selection methods were applied to reduce the issue of "over-fitting". The following methods were used:

- Variance Thresholding (as initial to remove repeating features i.e. features with 0 variance)
- Selecting K-best features in conjunction with χ^2

F. Models

A series of models were applied on the initial dataset containing all features. The models applied were the following:

- K-Nearest Neighbours
- Boosted Trees
- Bagged Trees
- Logistic Regression
- Support Vector Machine
- Random Forest
- Naive Bayes
- Gradient Boosting

G. Evaluation

The models were evaluated based on their precision, recall, accuracy and Receiver Operator Curve - Area Under the Curve (ROC-AUC) values. The receiver operator curve is a method to visualise and inspect models based on a curve of True Positive vs False Positive. In this context, we wish to minimise the false number of false negatives – the patient is predicted not having cancer, when they indeed have cancer. In this case, we favour recall, which measures this performance. Therefore, we also measure model performance on precision-recall curves.

Cross-validation ensured the shuffling, splitting, training, and testing of models. For the initial models, the F1 score – combination of precision and recall – was used to pick those with highest performance. The imbalanced nature of the dataset demands more attention to the number of positively predicted samples are actually positive. The metrics are defined where TP, FP, TN, FN are True Positive, False Positive, True Negative, False Negative respectively:

$$\begin{aligned}
\text{Accuracy} &= \frac{TP}{TP + TN + FP + FN} \\
\text{Recall/Sensitivity} &= \frac{TP}{TP + FN} \\
\text{Specificity} &= \frac{TN}{TN + FP} \\
\text{Precision} &= \frac{TP}{TP + FN} \\
\text{F1-Score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}$$

III. Results

A. Initial Inspection

Mean values of the classes for each feature showed no significant imbalances. However, LCa class tended to deviate much more. This could be due to the limited number of samples. It's notable that the LCa have high bounded values [Fig.4]. This highlights extremities of the class LCa, which can reduce model performance. Fitting an initial Logistic Regression model with the complete transformed data, $t(\mathbb{D})$, resulted in very good performance, where t is the natural logarithm. Thus, the data is likely to be highly linear.

The results for the initial logistic regression with L2 penalty, and penalising on the imbalanced class distribution:

Class	Precision	Recall	F1-Score	Accuracy
0	0.98	0.98	0.98	
1	0.90	0.91	0.91	
				0.97

The above results was performed with the splitting criteria aforementioned at CV = 10.

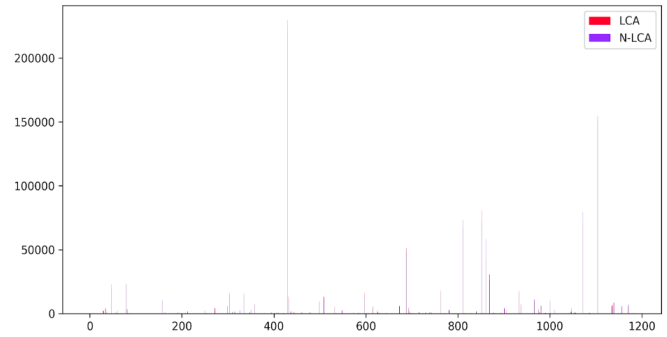

Fig. 1: The data split into its groups, LCa and Non-LCa. The means of the groups remain relatively similar to each other.

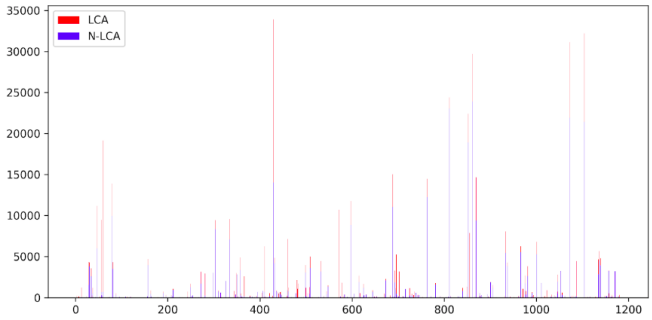

Fig. 2: Standard deviation of features in LCA and Non-LCA

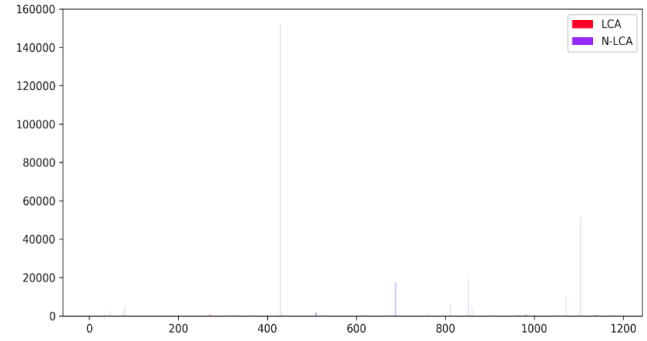

Fig. 3: Minimum values for each feature in LCA and Non-LCA

To ensure our model isn't over-fitting, reducing dimensionality can improve performance by picking uncorrelated independent features. To visualise any linear dependencies a heatmap of the correlation matrix highlights any linearly dependent features, where correlation is the Pearson correlation, defined as:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$$

ρ is the Pearson-correlation coefficient; values are between the range $\{-1, 1\}$.

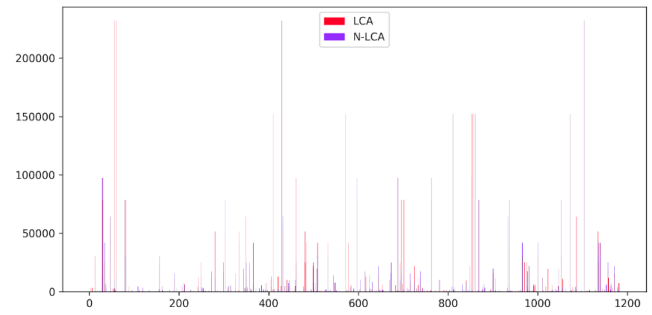

Fig. 4: Maximum values for each feature in LCa and Non-LCa

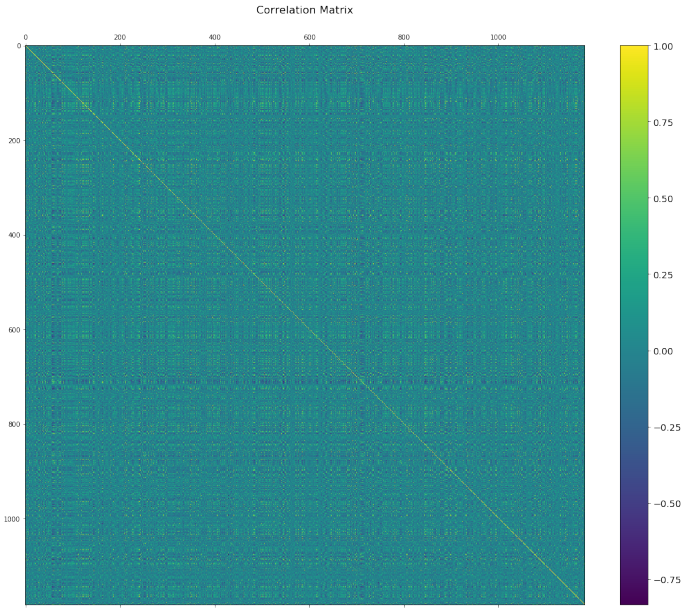

Fig. 5: Pearson-Correlation heatmap of the transformed data, $t(\mathbb{D})$.

Upon inspection of Fig.5, there are no clear indicators of high correlation. There are small bands of positively correlated samples, However not conclusive [Fig.6]. Visualising the data to determine visible clusterings in 2-D was achieved with PCA. The LCa points are seen scattered throughout, they however deviate slightly in the negative direction of PCA-2. Applying the t-SNE technique, clusters are evident, but due to our lack in complete understanding of the parameters, it is possible to be caught in a local optima [Fig.7].

B. Models

With the collection of our models, a grid-search coupled with cross-validation selected the best performing models based on the F1 score. Due to the large number of parameters and models, the top 10 models are displayed. All

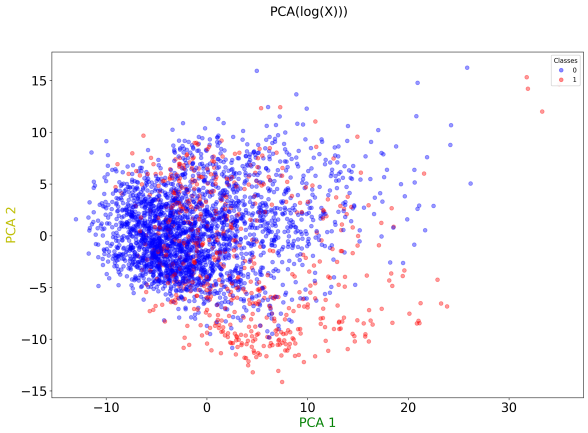

Fig. 6: PCA projection of $t(\mathbb{D})$ on to two dimensions.

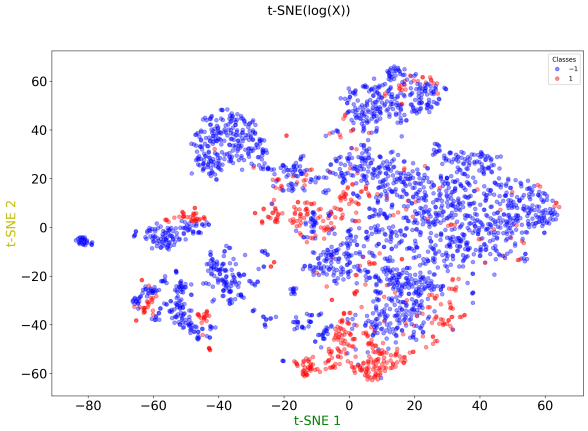

Fig. 7: t-SNE dimensionality reduction on to two dimesions. It is more likely to spread clusters.

supplementary data is available on our GitHub repository.

Model	Mean F1-Score	Params
SVM	0.929	[C=100,γ=0.001]
SVM	0.927	[C=100,γ=0.01]
SVM	0.927	[C=10,γ=0.01]
SVM	0.924	[C=1,γ=0.01
SVM	0.923	[C=10,γ=0.001]
LR	0.907	[C=1,penalty=L2]
SVM	0.880	[C=100,γ=0.0001]
LR	0.907	[C=0.1,penalty=L2]
SVM	0.823	[C=1,γ=0.001]
SVM	0.804	[C=10,γ=0.0001]

We chose the top model to compare with the Neural Network to see if Deep Learning can over-power the traditional methods. All models utilised over-sampling of the LCa class using Synthetic Minority Over-sampling Technique (SMOTE), synthesises new minority instances between existing (real) minority instances and CV = 5.

C. Neural Network

Two Neural Networks were designed. DeepNet2 used a heavy, high-node based architecture with 9 hidden layers with [200,400,600,800,1000,800,600,400,200], the input layer is the size of the data, and the output, 1. Each layer was activated with the Rectified Linear unit, ReLu, and the final output with the sigmoid activation function. L1 regularisation is applied on the first hidden layer. The optimisation procedure used was the Adam optimisation with a learning rate of 0.001 and decay of $\frac{learningrate}{epochs}$, number of epochs = 80, and the batch size at 64. The network can be visualised by the schematic below [Fig.8].

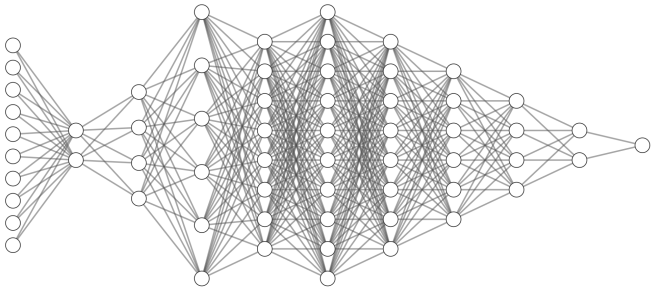

Fig. 8: DeepNet2 schematic. The layers are scaled by a factor of 10.

The second network, DeepNet3 was a much simpler network, with 4 hidden layers at, [number of features, 15, 10, 5] and output layer of 1. Activation of each layer was with a linear function, and L1 regularisation on the first layer. The output layer was equipped with the sigmoid activation function. The learning rate was set to 0.001, epochs was 150, batch size was 256, and the decay set to $\frac{learningrate}{epochs}$. All networks were scaled to $\{0, 1\}$ before training. Below is the schematic for DeepNet3 [Fig.10].

D. Evaluation

Evaluation was performed with the best classical models, with an additional sub-par model (Boosting) as a baseline, and the neural networks.

E. ROC-AUC Curves

ROC-AUC curves plot the FP rate against the TP rate. A good model sticks to the top left corner. All models perform well on the total dataset. Over-fitting could be a potential issue, and it is best to cross-validate and test models across multiple shuffle splits of the data.

The values of each model, with CV = 5.

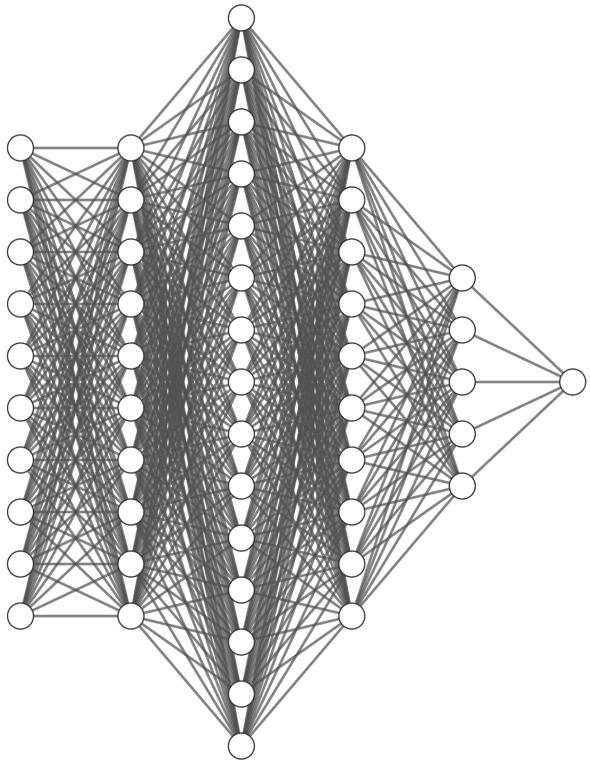

Fig. 9: DeepNet3 schematic. Input layer is scaled by a factor of 10.

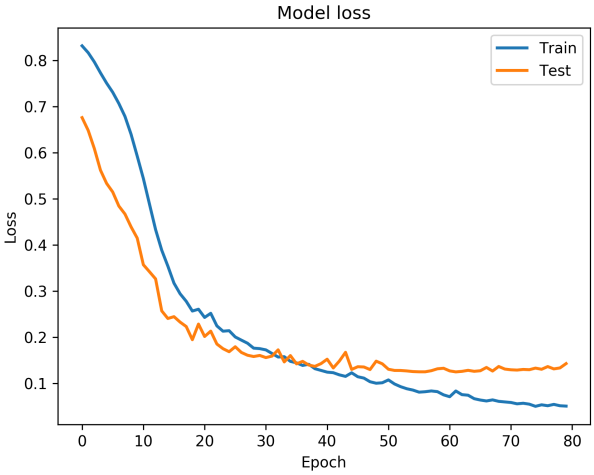

Fig. 10: Loss v epoch for DeepNet2

Model	Precision	Recall	Accuracy	AUC
DeepNet2	0.78	0.82	0.92	0.97
DeepNet3	0.87	0.92	0.96	0.99
LR	0.90	0.91	0.96	0.98
SVM	0.95	0.96	0.98	0.97

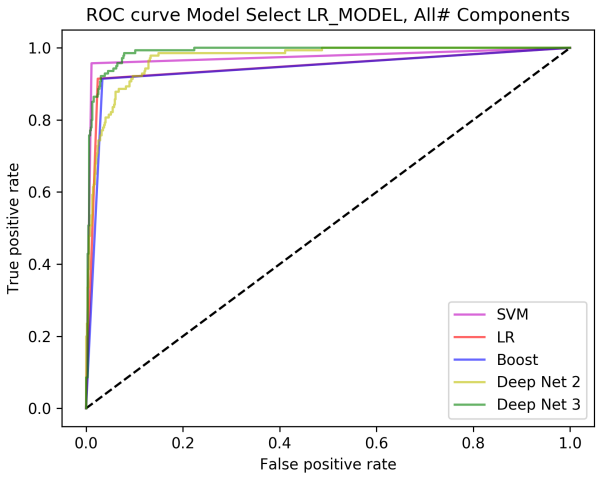

Fig. 11: All models on the ROC-AUC curve.

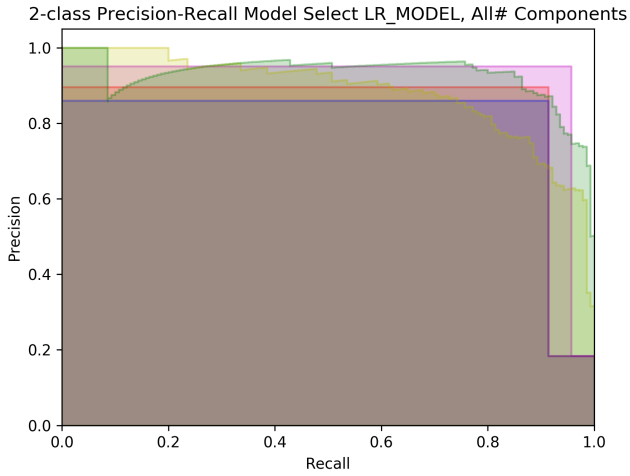

Fig. 12: All models with Recall-Precision curve.

Model	Time
DeepNet2	210
DeepNet3	40.5
LR	0.98
SVM	1.95

F. Combatting Over-fitting

Indeed, model performance could be greatly improved with feature selection. However, with feature selection techniques, reducing the number of features already drastically reduced performance. Using Selecting K-best from a χ^2 test with the features and class label produced the following results. With the features reduced down to a size of 300, performance greatly increases. The SVM model was held out in these cases to highlight the performance of LR against DeepNet3 and DeepNet2. Determining the optimal amount of features can be visualised based on the LR model. Here we use the coefficients fit to the model, and with the addition

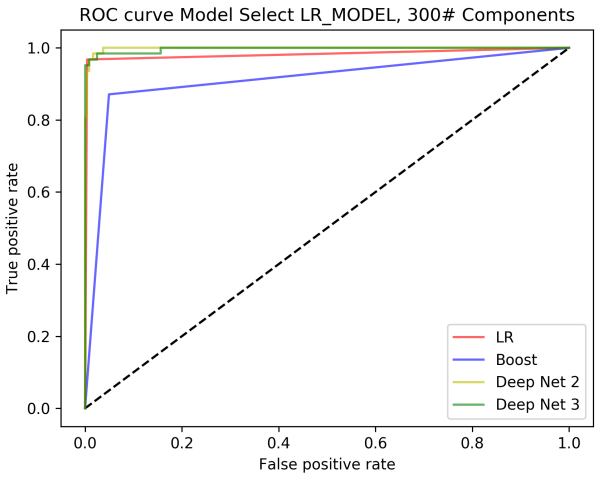

Fig. 13: ROC-AUC with 300 best features

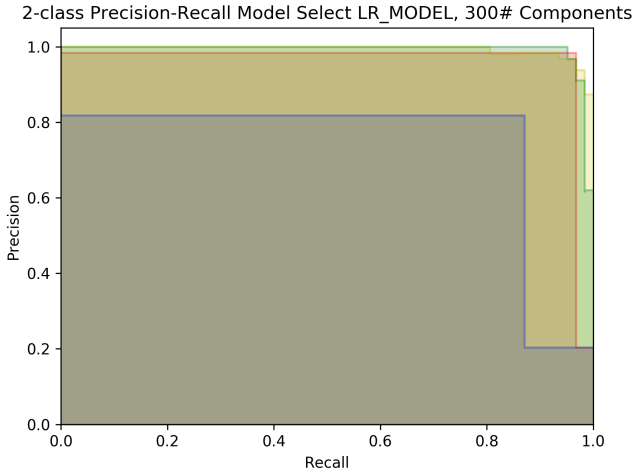

Fig. 14: Recall-Precision curve with 300 best features

of recursive feature elimination – the subsequent action of removing features from the full model and testing performance – only 244 components are needed to perform and plateau at 0.95% [Fig.15].

G. Induction

Despite features being highly dependent, is it necessary to perform feature selection at all? Although model performance increases slightly with LR and DeepNet2, an SVM trained on all of the data is already a strong classifier. The problem also lends itself from the limited size of the data. With a larger dataset, the dependencies and feature selection would prove functional. Thus, the inclusion of induction was necessary to solidify our claim. Removing features within the data proved to improve performance of the Neural networks, however damaged the classical model performance. Induction involves

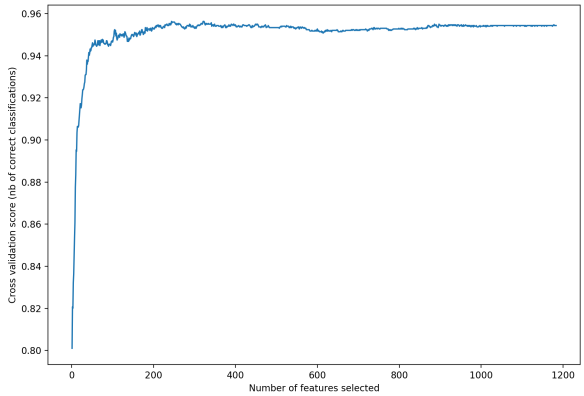

Fig. 15: Optimal number of components from LR with recursive feature elimination

inferring feature importance. Choosing best the best features then removing them cumulatively from the full model, then plotting number of components removed against performance difference.

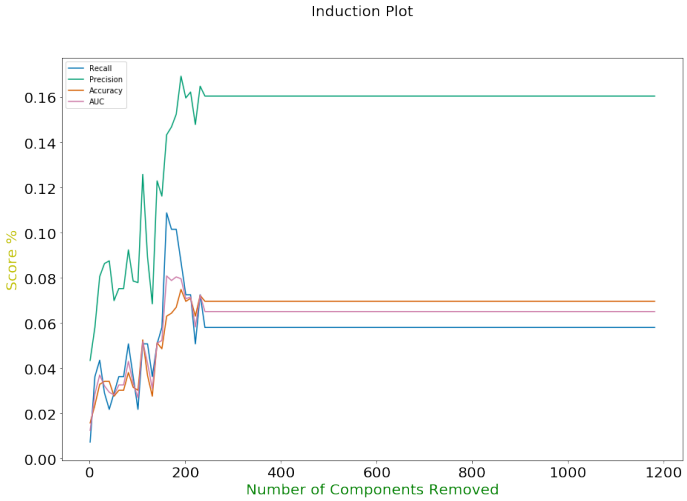

Fig. 16: Induction plot with the high performing SVM model. Removing best feature from χ^2 test. We see after roughly 440 best components removed, the model flattens, and there is no evident difference from the overall model and model features removed. The difference in performance is not extreme.

From [Fig.16] the overall change in performance across all measures isn't extreme. This implies there exists dependencies in the data, and from what we've gathered, very linear they are however not exactly linear.

H. Leave-One-Out Cross Validation

The data lacks in size, and hence using Leave-One-Out Cross Validation (LOOCV) can truly test a models

performance. Selecting the SVM, we performed LOOCV.

Class	Precision	Recall	F1-Score	Accuracy
0	0.98	0.99	0.99	
1	0.95	0.94	0.94	
				0.98

With an AUC score of 0.98.

IV. Conclusion

Finding a model in the use of medicine requires very special attention. Discovering a model with phenomenal accuracy does not justify its use in the real world. We have shown how traditional methods still have its place in Machine Learning, and play a major role in Science. Deep Learning techniques proved themselves to be robust to small, imbalanced datasets; learning a robust function given a limited samples. These methods require much more experience, time, and testing to optimise. We have proven the robust Support Vector Machine, equipped with a Radial-Basis Kernel outperforms modern, buzzword techniques. Due to our lack in experience in Deep Learning methods and completely grasping the algorithms, they did not shine as brightly as we wished.

It is also important to visualise results. Working in a inter-disciplinary environment is difficult to communicate mixed results from different backgrounds. Visualisation techniques aid in simplifying complex abstractions to drive a point home. With the SVM we are capable of communicating the decision boundaries of the model and highlighting potential areas of risk [Fig.17].

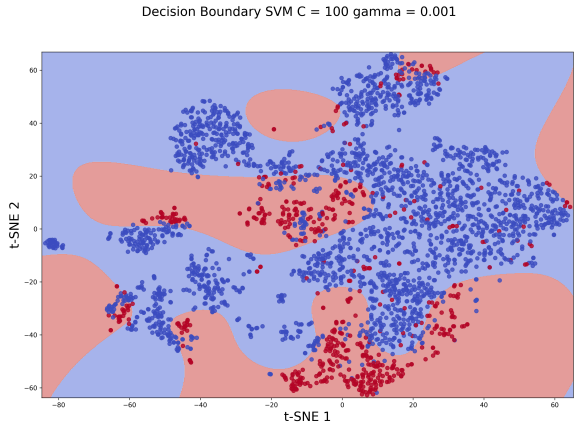

Fig. 17: t-SNE projection to 2D coordinates with the optimised SVM decision boundaries.

References

[1] C. Nardella, A. Lunardi, A. Patnaik, L. C. Cantley, and P. P. Pandolfi, "The APL paradigm and the "co-clinical trial" project," *Cancer Discovery*, 2011.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," 2017.

[3] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics of American 2019.," *CA: a cancer journal for clinicians*, 2019.

[4] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: Past, present and future," 2017.

[5] Y. Li and K. V. Kowdley, "Method for microRNA isolation from clinical serum samples," *Analytical Biochemistry*, 2012.

[6] A. Turchinovich, L. Weiz, A. Langheinz, and B. Burwinkel, "Characterization of extracellular circulating microRNA," *Nucleic Acids Research*, 2011.

[7] D. Ravi, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G. Z. Yang, "Deep Learning for Health Informatics," *IEEE Journal of Biomedical and Health Informatics*, 2017.

[8] K. Kapitanova and S. H. Son, "Machine learning basics," in *Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning*, 2012.