IT1244 Project Report: Cancer Detection Dataset

Team 22

Jishnu Appana

Manisekaran Harish

Aravindan Siddharth

D Paranitharan

Introduction

Cancer is a disease where patients remain asymptomatic for a long time before their symptoms become severe. At this point, treatment of the cancer is complicated and expensive. As a result, it is vital to detect cancer in the early stages to make the treatments as non-invasive as possible. Cancer involves genetic mutations or alterations which could change the length of DNA fragments hence changing the proportions of certain lengths of DNA fragments in the DNA molecule. Hence, by analysing the relationship between the proportion of known lengths of DNA fragments that occur in a person's DNA molecule, the stage or existence of cancer that patients are at can be identified.

There have been several works done in this field to detect cancer by analysing various patterns in the genome through machine learning. Some of these included

- Using Artificial Neural Networks and Decision Tree machine learning models to analyse patterns in certain markers of free DNA in the bloodstream to identify colorectal cancer
- Using Support Vector Machines (SVM) to classify data that was created through gel electrophoresis and PCR of gastric cancer samples (amplifying the amount of DNA present)
- Using Support Vector Classifier to detect prostate cancer tumour variants in cell-free DNA after identifying specific genetic markers that correlate to cancer.

However, all these methods only focused on a singular type of cancer. Moreover, the specific genetic markers and patterns correlating to cancer had to be identified beforehand. A more general approach was hence considered in this paper to build classifiers to classify a patient into whether they have cancer or not based on their DNA profile.

In addition to that, the classifiers should also classify (1) screening stage cancer vs healthy, (2) early-stage cancer vs healthy and lastly (3) cancer in general vs healthy.

Classifiers such as logistic regression, decision trees, Gaussian Naive Bayes classifier and K-nearest neighbours will be considered in this prediction analysis and their performance will be evaluated using metrics such as precision, recall and F1 score, as well as AUC (Area Under Curve).

Dataset

The data is separated into two data sets. One dataset contains the training data which has 2193 observations. The other dataset contains test data which has 1034 observations. There are 350 features, which comprise the different max normalized frequencies of DNA fragment lengths and the response variable, "class_label". There are 5 classes: healthy, early-stage cancer, screening-stage cancer, mid-stage cancer and late-stage cancer. If all 350 independent features are fitted into the model this may result in overfitting of the training models where the models generated perform very well with the training dataset but perform poorly with new unseen datasets such as the test dataset. Moreover, we aim to carry out the three classification models to carry out the three classifications above. Therefore, we have decided to preprocess the dataset as such before building out machine learning models:

Firstly, we check for any missing data points in the test and train datasets. Upon inspection, both datasets do not have any missing data points and are complete without any missing information. Secondly, we eliminate the non-screening stage cancer response variables for the (1) screening stage cancer vs healthy analysis, we eliminate the non-early stage cancer response variables for (2) early stage cancer vs healthy and we merge them moon healthy response variables into the cancer class for (3) cancer vs healthy. Thirdly, we standardise the dependent variables for the test and train datasets to ensure that all dependent variables in the data set contribute evenly to the output predicted. Next, we convert the response variable into binary form: 0 for the negative result (healthy) and 1 for the positive result (screening stage cancer/ early stage cancer/ cancer).

Subsequently, we perform feature selection via the following four methods to prune the dataset of unnecessary dependent variables that may cause overfitting as explained above.

- L1 Lasso regularisation: This technique adds a penalty term to the residual sum of squares calculated in a linear model. This therefore reduces the importance or

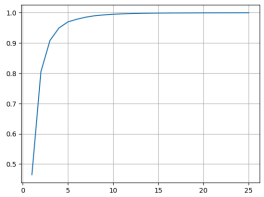
altogether removes certain features of the dataset that are deemed to be insignificant. We save these features under insignificant feature_1.

- Principal component analysis: This technique performs dimensionality reduction by reducing the number of features in this large dataset into principal components that retain most of the original information. We identify the number of features, n, that contribute most to variance. From the graph it could be read that drive features were sufficient to explain most of the variance in the dataset, hence making the other features irrelevant. We rank the features in their importance in explaining the variance and save the top n features under significant features_2 while the rest are saved under insignificant features_2.

- Recursive feature selection: For our analysis, we have applied recursive feature selection on the logistics regression model due to the binary nature of the output variables. The logistic regression model is progressively being trained on a subset of the features and iteratively removes the least important feature by comparing the above-mentioned hyperparameters such as p values. This will allow us to rank the features according to their importance in this logistic regression model. The top n features (that contribute to the most variance in the dataset) are saved under significant features_3 while the rest are saved under insignificant features_3.

- Correlation analysis: We use the dataset to form a 350 by 350 matrix of the correlation factors between the dependent features. We then mask the bottom triangular section of the matrix to prevent the repetition of correlation factors from occurring during analysis. We then compare these values with a threshold to identify the highly correlated variables. One of each pair of variables is randomly stored under insignificant features_4.

By assigning a score value to the features in the 4 sets of insignificant features, it could be decided that when a score value of 4 is required for removal, only 23, 18 and 24 features are preserved in the (1) screening stage cancer vs healthy, (2) early stage cancer vs healthy and lastly (3) cancer in general vs healthy analysis respectively.



Methods

We will split our technical approach into two to solve the problem. Firstly, we need to cluster the dataset so that we can properly cluster the dataset into different k groups. We use the k-means clustering algorithm for the first part of the problem

K-Nearest Neighbours

K-means clustering is an unsupervised learning method where similar data points are identified and grouped into clusters such that data points within each cluster are similar. It is an iterative process that first initialises k centroids randomly. Then, we will assign each data point to the closest centroid. After all data points have been assigned, within each cluster, we will calculate the new centroid of the assigned data points. Repeat the process until there is no change to the centroids. Once the algorithm has converged, the data points will have been grouped into k clusters based on their similarity to each other.

Logistic Regression

The logistic regression model uses a logistic function, given by the sigmoid function $g(x) = \frac{1}{1+e^{-x}}$ to model the relationship between the input features and the probability that the output variable is true. It returns a probability value between 0 and 1. The model is trained using a dataset that contains input features and output labels. During training, the algorithm adjusts the model parameters to minimise the error between the predicted probabilities and the actual output labels. Once the model is trained, it can be used to predict the output probability of new data. If the predicted probability is greater than 0.5, the output is predicted to be true, and if it is less than that, the output is predicted to be false.

Decision Trees (not covered in IT1244)

Decision trees are graphical representations of a series of decisions and their possible consequences. It consists of nodes, branches, and leaves, where each node represents a decision based on one or more input features, each branch represents a possible outcome of that decision, and each leaf node represents the final prediction or decision. The process of building a decision tree involves recursively splitting the data into smaller subsets based on the values of the input features, to maximise the information gain or minimise the impurity at each split. One advantage of decision trees is that they are easy to interpret and visualise. However, decision trees are prone to overfitting, especially when the data has a large number of features. Using methods such as gradient boosting can help mitigate the issue (Analytixlabs, 2022).

To implement the decision tree, we used 'RandomForestClassifier', "AdaBoostClassifier", and "GradientBoostingClassifier",

from the module 'tree' in 'scikit-learn'. We went for a max_depth value of 8 (which is defined by the number of layers between the root node and the leaf node), as we wanted our model to converge.

Gaussian Naive Bayes (not covered in IT1244)

The Gaussian Naive Bayes classifier is the modified version of the general Naive Bayes classifier. It is based on the Bayes' Theorem such that

$$P(y|X) = \frac{P(X|y).P(X)}{P(y)}$$

X represents the list of independent predictors where X = $x_1 + x_2 + \ldots + x_n$. y is the class label and the conditional probability is the probability that y occurs given the predictors X. A more comprehensive version is as shown below.

$$P(y|x1, x2, x3..xN) = \frac{P(x1|y).P(x2|y).P(x3|y)\ldots P(xN|y).P(y)}{P(x1).P(x2).P(x3)\ldots P(xN)}$$

However, the general Naive Bayes classifier only allows for categorical input variables. So the Gaussian Naive Bayes classifier had to be adopted instead since the input variables in this dataset are numerical and continuous. The main assumption we make with the selection of this model is that each input variable (max normalised frequency of DNA fragment length X) follows a normal distribution. When doing exploratory data analysis, we realised that most of the input variables did not follow a Gaussian distribution hence we took the log-values of all the input variables so that the distribution is closer to that of a normal distribution.

Results & Discussions

Firstly, let us discuss the results of the K-means clustering algorithm. We used classification reports of each classifier to get metrics such as f1-score, recall, precision and AUC.

$$Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN}$$
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

We tried not to use accuracy as it only takes into account the true positives (TP) and true negatives (TN), which is not what we are most concerned about. The cost of having a false negative (FP) is higher than a false positive (FN) as we can classify a healthy person to be a cancer patient and that will just result in them going for checkups. However, we cannot classify a cancer patient as healthy as they might not seek treatment thinking that they are healthy. Therefore, we should seek to minimise the number of FN, and consequently, maximise the precision. The higher the precision, the better. Let's analyse the metrics of each method.

Logistic Regression

	precision	recall	f1-score	support
0	0.62	0.08	0.14	99
1	0.64	0.97	0.77	165
accuracy			0.64	264
macro avg	0.63	0.53	0.46	264
weighted avg	0.63	0.64	0.53	264

The above is the classification report for our logistic regression model. It has mediocre precisions of below 0.7 for both outcomes. It has a high recall score for classifying someone as a cancer patient but extremely low for the other outcomes. The precision-recall curve of the model classifying healthy and cancer patients is shown below. As seen, it has a high AUC value and f1-score.

	precision	recall	f1-score	support
0	0.99	0.48	0.64	32175
1	0.86	1.00	0.92	99393
accuracy			0.87	131568
macro avg	0.92	0.74	0.78	131568
weighted avg	0.89	0.87	0.85	131568

The classification report for the Random Forest Classifier is much better. The precision for both outcomes is close to 1.

	precision	recall	f1-score	support
0	0.88	0.84	0.86	32175
1	0.95	0.96	0.96	99393
accuracy			0.93	131568
macro avg	0.92	0.90	0.91	131568
weighted avg	0.93	0.93	0.93	131568

Similarly, the precision of XGBoost is quite solid with values close to 1 as well.

We then plotted various ROC-AUC curves that tell us how efficient each model is.

This is the precision-recall curve of the model classifying healthy and cancer patients. As seen, it has a high AUC value and f1-score. The classification report for this classifier also shows that it has a high recall and decent precision as well.

Decision Tree

Classification Report:

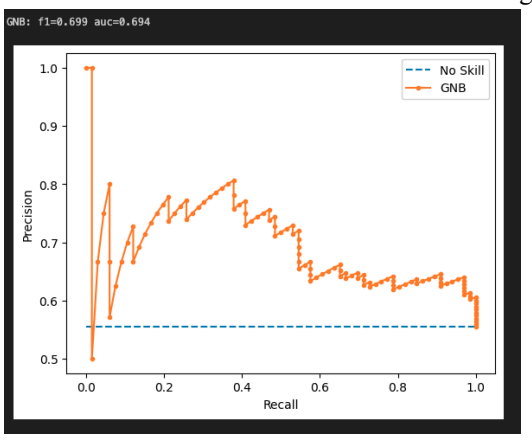
	precision	recall	f1-score	support
Cancer	0.98	0.92	0.95	993
Healthy	0.21	0.49	0.29	41
accuracy			0.91	1034
macro avg	0.59	0.71	0.62	1034
weighted avg	0.95	0.91	0.92	1034

This is the best-fitting Decision Tree comparing the 3 different methods of boosting.
As the decision tree is easily overfitted, 3 methods of enhancing results were used ADAboosting, Gradient boosting and Randomforest Classifier. As we want to reduce the FNR we picked the best tree that gave the best f1 score. This is because a high precision score or a high recall score results in an overfitted tree. Hence, the f1 score was used as a balanced score of precision and recall.

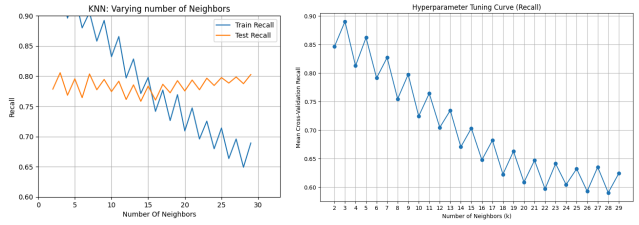
Gaussian Naive Bayes (GNB)

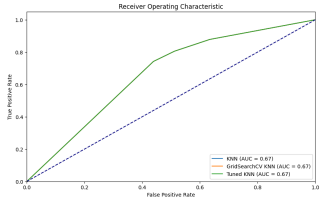
	precision	recall	f1-score	support
0	0.62	0.45	0.52	53
1	0.64	0.77	0.70	66
accuracy			0.63	119
macro avg	0.63	0.61	0.61	119
weighted avg	0.63	0.63	0.62	119

The above is the classification report for our GNB model. It has mediocre precisions of below 0.7 for both outcomes as well. It has a high recall score for classifying someone as a cancer patient but extremely low for the other outcomes. The precision-recall curve of the model classifying healthy and cancer patients is shown below. This is not the best model as even AUC and f1-scores are not that high.



KNN





	precision	recall	f1-score	support
Healthy	0.08	0.44	0.14	41
Cancer	0.97	0.80	0.88	993
accuracy			0.79	1034
macro avg	0.53	0.62	0.51	1034
weighted avg	0.94	0.79	0.85	1034

Above is the performance of the KNN model for Healthy vs cancer as the value of K changes. To find the best K for the best recall score Hyperparameter tuning was done to find the best combination of parameters to give the best results. The best K results were verified using GridSearchcv which uses cross-validation to find the identify the best combination of parameters and K values.

References

https://www.sciencedirect.com/science/article/pii/S2001037014000464#:~:text=A%20variety%20of%20these%20techtechniques,effective%20and%20accurate%20decision%20making. Tanos, R., Tosato, G., Otandault, A., Al Amir Dache, Z., Pique Lasorsa, L., Tousch, G., ... & Thierry, A. R. (2020). Machine learning‐assisted evaluation of circulating DNA quantitative analysis for cancer screening. *Advanced Science*, *7*(18), 2000486.

Kong, W., Tham, L., Wong, K. Y., & Tan, P. (2004, January). Support Vector Machine Approach for Cancer Detection Using Amplified Fragment Length Polymorphism (AFLP) Screening Method.. In *ACM International Conference Proceeding Series* (Vol. 55, pp. 63-66).

Cario, C. L., Chen, E., Leong, L., Emami, N. C., Lopez, K., Tenggara, I., ... & Witte, J. S. (2020). A machine learning approach to optimizing cell-free DNA sequencing panels: with an application to prostate cancer. *BMC cancer*, *20*, 1-9

https://www.ibm.com/topics/lasso-regression#:~:text=Lasso%20regression%E2%80%94also%20known%20as,W%20)%20%2B%20%7C%7Cw%7C%7C1

https://www.ibm.com/topics/principal-component-analysis#:~:text=Principal%20component%20analysis%20(PCA)%20reduces,of%20variables%2C%20called%20principal%20components.

https://medium.com/@rithpansanga/logistic-regression-for-feature-selection-selecting-the-right-features-for-your-model-410ca093c5e0#:~:text=Recursive%20Feature%20Elimination%20(RFE)%20is,the%20desired%20number%20of%20features.