Microarray Data Analysis (cancer detection)

Mohamed Essam, Mohamed Maher, Mohamed Fekry, Mohamed Magdy, Malek Ashraf, Samaa Rabea

Faculty of computers and Informatics, Zagazig University, Zagazig, 44519, Egypt
Emails: mosalah36000@gmail.com; mahermo656@gmail.com; mofikry17@gmail.com; mohamedmagdy1101@gmail.com; malekashraf1500@gmail.com; samaarabea7@gmail.com
*Correspondence: mosalah36000@gmail.com

Abstract

Microarray data is an increasingly important tool for providing information on gene expression for analysis and interpretation. Researchers attempt to utilize the smallest possible set of relevant gene expression profiles in most gene expression studies to enhance tumor identification accuracy. This research aims to analyze and predicts cancer data employing a machine learning approach and feature selection technique based on a Univariate feature selection and GridSearch method to build the optimal classifier. With the aim of increasing the prediction model's accuracy level. So that it expectedly helps us to exactly predict and diagnose cancer. The development of microarray technology has supplied a large volume of data to many fields. It has been applied to prediction and diagnosis of cancer, so that it expectedly helps us to exactly predict and diagnose cancer. To precisely classify cancer, we must select genes related to cancer because extracted genes from microarray have many noises. In this paper, we attempt to explore many features and classifiers.

Keywords – Microarray data- Feature selection- Machine learning- GridSearch- Cross validation –Cancer

1. Introduction

Cancer is one of the deadliest diseases in the world and it is a subject of concern because until till date they cannot found the real treatment for this disease. As the cancer is a disease involving dynamic genome changes the considerable efforts have been made by researchers and technologists to explore the precise assessment and diagnose of the cancer, including the tumor prediction. If and only if this disease is detected in early stage, patients having this disease can be saved. If it is detected in latter stage, then chance of survival will be very less. Because of this, early and true diagnosis is an important issue and plays a key role to cure this disease. developed a generic cancer classification approach based on DNA microarray gene expression monitoring. They also proposed that such microarrays might provide a classification tool for cancer. Microarray based gene expression has been widely. Early detection of cancer is very important for proper diagnosis and treatment. Microarray dataset consists of thousands of genes and the number of samples is usually small. It is a challenging task to identify the most relevant genes from such types of microarray data as not all genes have sufficient follow-up-information and many of them are redundant. feature selection is current method of obtaining feature genes for cancer classification-based gene expression data. feature selection methods do not create a new subset of features. They work by removing

non-relevant or redundant features and retains the best classification accuracy. Feature selection does not involve transformation of the original features thus decrease the dimensionality problem and builds a robust learning model from the selected data. Therefore, the methods of feature selection have gained further interest.

1.1Objectives:

* Building a machine learning code for detection of Five cancers based on microarray dataset using machine learning classification techniques.

*This project aims to analyze and employing a machine learning approach and feature selection technique (Univariate feature selection) [1].
* We aim to save doctors time and effort; the evaluation Techniques aims to helping doctors and pathologists to detect cancer at an early stage where they can deal with it earlier, hoping to save many lives.

1.2 Mythology / Outline

Figure. 1 shows this study's methodology. The process starts with data collection. Starting with splitting the data using cross-validation then use it with GridSearch method to get the best parameters possible and save them for later. On the other side split the data using train-test technique then apply feature selection to get

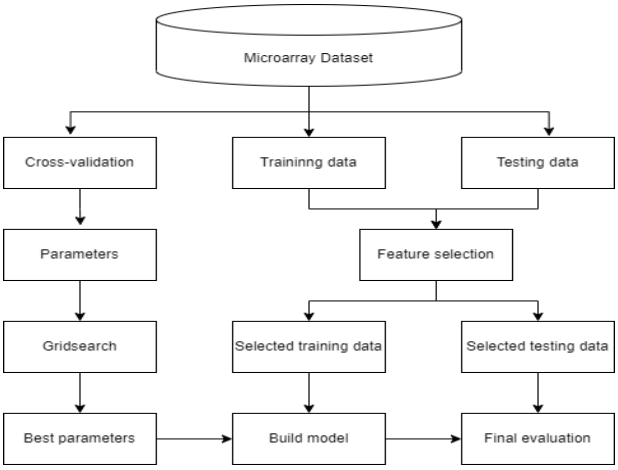


Figure. 1

new selected data. After all of these done now, we can build the model using the parameters from GridSearch and train it with the selected training data. Finally use selected testing data to evaluate the model accuracy.

1.2.1 Phase 1 data acquisition

Cancer gene expression data has been obtained from Gene Expression Omnibus (GEO) and Alon (1999) [2]

Type	Genes	Classification type	No. of samples	
Colon	2000	Normal	62	22
		Tumor		40
Breast	54676	Normal	116	15
		Tumor		101
Leukemia	47323	Normal	45	9
		Tumor		36
Liver	54676	Normal	114	58
		Tumor		56
Lung	54676	Normal	165	50
		Tumor		115

Figure. 2

The datasets distribution shown in Figure.2, we have a vary in the shape and samples of the datasets.

1.2.2 Phase 2 evaluation of classification without feature selection

In this phase, We Used GridSearch [3] on 3 different classifiers [KNN, SVC, Random Forest] with CV tenfold cross-validation in order to get the best parameters to build our models with.

1.2.3 phase 3 perform feature selection

Feature selection was a must to get the best result and overcome the overfit and underfit.

In this project we used Univariate feature selection since it's not commonly used.

1.2.4 phase 4 build the final model

In the last phase we used the parameters we got from GridSearch in phase 2 to build the models and trained them

2. LITERATURE SURVEY:

Related works

Recently, a lot of research has been developed to work on healthcare data by incorporating machine learning techniques with feature selection methods. Park & Kim developed a model with 20 datasets of microarray gene expressions to examine the property of the model based on sequential random k-nearest neighbor feature selection method [4]. An intelligent technique based on feature selection using t-statistic was proposed for colon cancer prediction. Authors achieved almost 85% accuracy using t-statistic feature selection method and Support Vector Machine (SVM) classifier [5]. A Fuzzy Decision Tree (FDT)-based feature selection algorithm was introduced by S.A. Ludwig et al. [6] to analyze gene expression for colon cancer data classification and achieved 80.28% accuracy by selecting 20 features. Modified Analytic Hierarchy Process (MAHP) with Probabilistic Neural Network (PNN) was introduced [7] in as a novel aggregate gene selection method for microarray data classification. The experimental results demonstrated that the proposed MAHP method obtained the top accuracy of 88.89% for colon cancer diagnosis with a benefit of inexpensive computational cost. Authors [8] used Fast Correlation Based Feature Selection (FCBFS) method with SVM as optimized by Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) to improve cancer classification quality. They observed that the classification model based on PSO, and ABC attained 93.55% classification accuracy for colon cancer prognosis. Mao long et al. [9] developed a Binary Quantum-Behaved Particle Swarm Optimization (BQPSO) and SVM with leave-one-out cross validation (LOOCV) based method for cancer feature selection and classification. They concluded that the proposed algorithm produced the classification results, with best accuracy of 93.55% and mean accuracy of 92.52%

for colon cancer datasets. Authors [10] relies on the methodology that uses Information Gain (IG) for feature selection, Genetic Algorithm (GA) for feature reduction, and Genetic Programming (GP) for cancer classification based on the gene expression profiles. For colon tumor classification, the suggested algorithm achieved an accuracy of 85.48%. A method of selecting features using Genetic Algorithm (GA) was proposed to select the best subset of features for breast cancer diagnosis system [11]. Random forest is an ensemble-based classifier consisting of a collection of trees of classification and regression (CART). Compared to other classifiers like Adaboost, SVM, neural network, decision tree, it reduces overfitting and therefore is more accurate. It is also used as a feature selection approach to rank the feature importance.

For experimental testing, we have considered each of the 2000 genes to classify the whole dataset into two classes: normal and abnormal. Table 4 shows the confusion matrix and the performance analysis with respect to recall, precision, F1-measure, and accuracy scores across the two different classes is shown in Table 5. As can be seen in Tables 4 and 5, the results of our classification model based on random forest that can correctly detect 52 items out of a total of 62 items, resulting in a weighted recall, precision, and F1-score of 83.68%, 83.87%, and 83.68% respectively. The overall accuracy of

Table 4 Confusion matrix of the model without feature selection

Actual class	Predicted class	
	Abnormal	Normal
Abnormal	36	4
Normal	6	16

Table 5 Performance analysis of the model without feature selection

Class	Recall	Precision	F1-score	Accuracy (%)
Abnormal	0.85714	0.90	0.8780	
Normal	0.80	0.72727	0.7619	83.871
Weighted measure (%)	83.68	83.87	83.68	

evaluation of classification without feature selection:

In this phase, a RF classifier with tenfold cross-validation was performed with all the attributes to evaluate the performance of the model

Classification algorithm description

In this study, a renowned classification algorithm for the prediction model namely random forest was evaluated in the prediction of colon cancer. RF is a combined classifier formed by combining a collection of unpruned decision trees, i.e., CART (classification and regression trees). A detailed overview of CART procedure can be found in Chang and Wang [12] and Harb et al. [13] The RF prediction when conducting classification analysis is the unweighted majority of individual trees class votes.

Feature selection techniques

Improving the accuracy of predictions by identifying certain features on the grounds of correlation statistics is known as feature selection, since this will represent a very good rate of classification. On the other hand, the classification process is the way to present out the test accuracy of the result. It is also possible, using this technique, to assess accuracy as a function of the ratio of predicted samples to total samples

Table 1 had listed 32 different approaches of applying the hybrid feature selection method, 4 of these methods had achieved a better classification accuracy of 90% or above. Most of the state-of-the-art technologies found that for the colon cancer dataset, the mRMR, GA, IG, and PSO are commonly applied for the hybrid feature selection and evaluates to better result

Table1 [14]

Method		Accuracy
Feature Selection	Classifier	[%]
PSO+GA	SVM	91.90
mRMR + PSO	SVM	90.32
Genetic Algorithm (GA)	SVM	90.32
CFS + Wrapper (J48)	SVM	89.03
Filter (F-Score+IG) + Wrapper (SBE)	SVM	87.50
CFS + Wrapper (Random Forest)	SVM	87.10
CFS + Wrapper (Random Trees)	SVM	85.48
mRMR	SVM	85.48
mRMR+GA-SVM	SVM	85.48
mRMR+GA	SVM	85.48
FSBRR + MI	KNN	91.91
CFS + Wrapper (Random Forest)	KNN	87.10
CFS + Wrapper (J48)	KNN	85.48
CFS + Wrapper (Random Trees)	KNN	82.26
Genetic Algorithm (GA)	DT	88.8
PSO+GA	DT	83.9
GE Hybrid	DT	83.41
IG	DT	77.26
MF-GE	DT	76.64
PSO+GA	Naïve Bayes	85.50
GE Hybrid	Naïve Bayes	84.96
MF-GE	Naïve Bayes	75.07
mRMR	Naïve Bayes	66.13
MIM+AGA	Extreme Learning Machine (ELM)	89.09
Information Gain (IG) & Standard Genetic Algorithm (SGA)	Genetic Programming	85.48
GE Hybrid	7-Nearest Neighbor	85.34
MF-GE	7-Nearest Neighbor	68.78
GE Hybrid	3-Nearest Neighbor	84.93
MF-GE	3-Nearest Neighbor	77.01
GE Hybrid	Random Forests	81.67
MF-GE	Random Forests	74.35
PCA	GA + ANN	83.33

2.2. The limitations of previous studies

In the light of above, the limitations of previous studies are highlighted below

• The most literatures reported good results when they limited the quantity of gene selection to a fixed number of genes prior to classification, thus ignoring the rest of genes which may cause an ignore to important gene.

• Many studies had claimed that reducing the number of genes will enhance the classification accuracy, but as shown in Table 1 the superlative accuracy reached 92%. Thus, there is a need to a better method or a framework model to proof the classification enhancement of the hybrid methods.

• To the superlative of the author's knowledge, there is no previous study stated in the literature had touched the hybrid feature selection method with the approach of a two-stage multifilter hybrid selection method.

3. PROBLEM SPECIFICATION

Our project idea was at first detection of colon cancer only because Over time, there are many diseases, especially cancers, that continue to negatively affect people's lives, and regarding these diseases, we have taken into account an integral part of these cancers which is colon cancer.

Why was Colon cancer being our interest?

Early detection of colon cancer is very important for proper diagnosis and treatment. Microarray dataset consists of thousands of genes and the number of samples is usually small. It is a challenging task to identify the most relevant genes from such types of microarray data as not all genes have sufficient follow-up-information and many of them are redundant.

The original dataset

Alon et al. (1999) [2] have presented a data set that contains gene expression levels of 40 tumor and 22 normal colon tissues for 6500 human genes obtained with an Affymetrix oligonucleotide array. complementary to more than 2000 human genes and expressed sequence tags.

Alongside Colon cancer we tested our mythology on 4 other datasets for different type of cancer was published by structural bioinformatics and computational biology (SCBC) based on (CuMiDa: An Extensively Curated Microarray Database for Benchmarking and Testing of Machine Learning Approaches in Cancer Research) article. [15]

The other datasets are:

Leukemia_GSE71935[16]: Gene expression profiling in 36 JMML patients and 9 healthy donors (Validation cohort)

Lung_GSE19804[17]: RNA was extracted from paired tumor and normal tissues for gene expression analysis. But the actual data size after the preprocessing 50 Normal and 115 Tumor.

Breast_GSE42568[18]: Gene expression profiling of 101 breast cancer and 15 normal breast biopsies.

Liver_GSE76427[19]: Microarray expression data for 56 tumor and 58 adjacent non-tumor tissues from hepatocellular carcinoma patients.

The method used!!

Feature selection is the current method we used to obtain distinct genes for cancer grading based gene expression data. There is another way is Feature transformation is a process in which to create new set of features from original features to achieve the purpose of feature reduction.

Feature selection is a technique where we choose those features in our data that contribute most to the target variable. In other words, we choose the best predictors for the target variable.

Univariate feature selection works by selecting the best features based on univariate statistical tests

Using UVS[1] we have 3 different options for classification as score function which are: chi2, f_classif, mutual_info_classif.

We tested the 3 and choose the best and the most commonly used which is chi2.

Chi2 Compute chi-squared [20] stats between each non-negative feature and class. This score can be used to select the number features with the highest values for the test chi-squared statistic from data X, which must contain only non-negative features such as Booleans or frequencies relative to the classes.

Models Used

Since the main problem we solve is mainly based on machine learning, we used these models divided into two categories, supervised classification models to analyze and evaluate the data. Supervised Classification Models: Random Forest Classifier, KNeighbors Classifier and Support Vector Classifier.

Models to analysis and evaluate data and models: GridSearchCV[3], Train Test Split, KFolf, and CrossValidation.

4.implementation

4.1 Machine learning model

Modules for data preparation and visualization

We used some libraries to prepare and visualize the data like [pandas, seaborn, matplotlib]

Modules supervised Classification models:

we used some classifier from sklearn [RandomForestClassifier, SVC, KNeighborsClassifier] data analysis and model evaluation

Modules for data analysis and model evaluation

We used some evaluation techniques from sklearn [GridSearchCV, train_test_split, KFold, classification_report, OrdinalEncoder]

Data preparation:

We kept the data preparation as simple as possible since we will be using feature selection techniques later on. We just dropped and null values from the datasets and removed duplicates sample.

K-fold cross-validation description [21]

Cross-validation is a resampling procedure used to evaluate machine-learning models on a limited data sample.

The method has a single parameter called k which corresponds to the number of groups to be divided into a given data sample. Therefore, the technique is often referred to as k-fold cross-validation. When the specific value of k is chosen to be 10 then the model is called tenfold cross-validation.

K-fold cross-validation is carried out according to the following steps:

• Spilt the whole dataset into k equal parts where each spilt of the data is called a fold. Let f1, f2,……fk be the name of each fold

• for i=1 to k

o Keep the fold fi

as a validation set and the remaining k-1 folds in the training set.

o Fit a model on the training set and evaluate the accuracy of the model on the validation set.

• Calculate the model's accuracy by averaging the accuracy of all k-fold cross-validation cases

Fine -Tune the Model [22]

A Machine Learning model is defined as a mathematical model with several parameters that need to be learned from the data. By training a model with existing data, we can fit the model parameters.

However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

GridSearchCV

RandomizedSearchCV

Grid search [3]

The overall idea of the grid search is to create a grid of all possible hyperparameter

combinations and train the model using each one of them. Hyperparameters are the

external characteristic of the model, can be considered the model's settings, and are

not estimated based on data-like model parameters. These hyperparameters are

tuned during grid search to achieve better model performance.

In this section we tried 3 different type of classifiers which are SVC, Random Forest and Knn to get their best parameters with our dataset.

we build the grid search model with tenfold cross-validation and scoring is accuracy

Drawback: GridSearchCV will go through all the intermediate combinations of hyperparameters which makes grid search computationally very expensive.

SVM also has some hyper-parameters (like what C or gamma values to use) and finding optimal hyper-parameter is a very hard task to solve. But it can be found by just trying all combinations and see what parameters work best.

Each one we used hyperparameter in GridSearchCV was different, and everyone gave excellent results

Random Forest: [23]

Random forest is a supervised learning approach used in machine learning for classification and regression. It's a classifier that averages the results of many decision trees applied to distinct subsets of a dataset to improve the dataset's projected accuracy.

It's also known as a meta-estimator since it fits a number of decision trees on different sub-samples of datasets and utilizes the average to enhance the model's forecast accuracy and prevent over-fitting. The size of the sub-

sample is always the same as the size of the original input sample, but the samples are generated using replacement. It produces a "forest" out of a collection of decision trees that are frequently trained using the "bagging" method. The main idea of the bagging approach is that combining many learning models enhances the final result. Rather than relying on a single decision tree, the random forest gathers forecasts from each tree and predicts the ultimate output based on the majority of votes.

Random forest algorithm description: [24]

For the original dataset D (X, Y), RF constructs the basic decision trees:

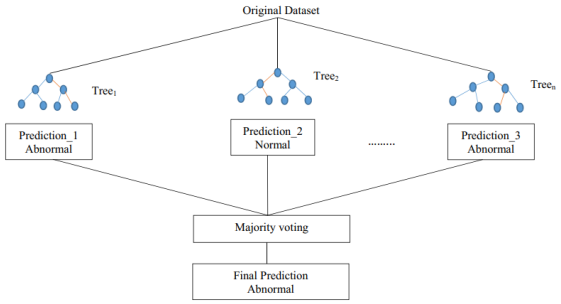


Figure.3

D(X, Y) = (3) {(x1, y1) , (x2, y2) ……(xn, yn)}

where, n is the number of training observations consists of a set of instances whose class membership is

known, K is the number of class and (xi, yi) ϵ (X, Y). Find

an optimal classifier hK(X) that minimizes the error with respect to the original dataset, then the combined classifier can be described as:

h = (4) {h1(X), h2(X)……hK (X)}

K-Nearest Neighbors: [25]

K nearest neighbors is a straightforward method that maintains all existing examples and categorizes new ones using a similarity metric (e.g., distance functions).

KNN has been utilized as a non-parametric approach in statistical estimates and pattern recognition since the early 1970s. It's a form of lazy learning since it doesn't try to build a generic internal model; instead, it only saves instances of the training data. The classification is determined by a simple majority vote of each point's k closest neighbors.

A case is categorized by a majority vote of its neighbors, with the case being allocated to the class having the most members among its K closest neighbors as determined by a distance function. If K = 1, the case is simply allocated to the nearest neighbor's class.

K-Nearest Neighbors algorithm description: [26]

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables, the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

How to choose the value of K?

'k' in KNN is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process. K actually is the number of neighbors considered. Thus, when fitting a model with k=2, the two closest neighbors are used to smooth the estimate at a given point. It's an extremely important parameter, and multi scale ensembles of KNN show promise on a variety of regression problems. If the value of K is 3, it means that the 3 nearest neighbors are considered for computation.

K-NN Algorithm is based on feature similarity: choosing the right value of k is a process called parameter tuning and is important for better accuracy.

Few ideas on picking a value for 'K'

There is no structured method to find the best value for "K". We need to find out with various values by trial and error and assuming that training data is unknown.

Choosing smaller values for K can be noisy an

will have a higher influence on the result.

Larger values of K will have smoother decision boundaries which mean lower variance but increased bias. Also, computationally expense.

Support Vector: [27]

Machine learning with maximization (support) of separating margin (vector), called support vector machine (SVM) learning, is a powerful classification tool that has been used for cancer genomic classification or subtyping. Today, as advancements in high-throughput technologies lead to production of large amounts of genomic and epigenomic data, the classification feature of SVMs is expanding its use in cancer genomics, leading

to the discovery of new biomarkers, new drug targets, and a better understanding of cancer driver genes. Herein we reviewed the recent progress of SVMs in cancer genomic studies. We intend to comprehend the strength of the SVM learning and its future perspective in cancer genomic applications.

SVM is a powerful method for building a classifier. It aims to create a decision boundary between two classes that enables the prediction of labels from one or more feature vectors. This decision boundary, known as the hyperplane, is orientated in such a way that it is as far as possible from the closest data points from each of the classes. These closest points are called support vectors' learning is one of many ML methods. Compared to the other ML methods SVM is very powerful at recognizing subtle patterns in complex datasets. Cancer is a genetic disease where the genomic feature patterns or feature function patterns may represent the cancer subtypes, the outcome prognosis, drug benefit prediction, tumorigenesis drivers, or a tumor-specific biological process. Therefore, the Artificial Intelligence of SVM can help us in recognizing these patterns in a variety of applications.

Support Vector algorithm description: [28]

SVM is a powerful method for building a classifier. It aims to create a decision boundary between two classes that enables the prediction of labels from one or more feature vectors. This decision boundary, known as the hyperplane, is orientated in such a way that it is as far as possible from the closest data points from each of the classes. These closest points are called support vectors. Given a labeled training dataset:

(x1, y1), ..., (xn, yn), xi ∈ Rd and yi ∈ (−1, +1)

where xi is a feature vector representation and yi the class label (negative or positive) of a training compound i.The optimal hyperplane can then be defined as: wxT + b=0

where w is the weight vector, x is the input feature vector, and b is the bias. The w and b would satisfy the following inequalities for all elements of the training set:

wxiT + b ≥ +1 if yi=1

wxiT + b ≤ −1 if yi=–1

The objective of training an SVM model is to find the w and b so that the hyperplane separates the data and maximizes the margin 1 / || w ||2

Train-test split and evaluation metrics.

It is a good idea to partition the original dataset into training and test sets. The test set is a sample of the data that we hold back from our analysis and modeling. We use it at the end of our project to confirm the accuracy of our final model. It is the final test that gives us confidence in our estimates of accuracy on unseen data. We will use different size of the data- set for model training and different for testing since we have 5 data every data has its unique Train-test split size, and this will be shown later.

Base models:

To check if our mythology really works, we built the base models for every dataset and compare it with the model we will be building after to see if there's any real evaluation we managed to achieve. So, we kept it simple with base models and the data before feature selection

Feature selection technique [1]

Improving the accuracy of predictions by identifying certain features on the grounds of correlation statistics is known as 'feature selection'., since this will represent a very good rate of classification. On the other hand, the classification process is the way to present out the test accuracy of the result. It is also possible, using this technique, to assess accuracy as a function of the ratio of predicted samples to total samples.

Feature selection we use is Univariate feature selection

Univariate feature selection [1] works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. Scikit-learn exposes feature selection routines as objects that implement the transform method:

SelectKBest removes all but the k highest scoring features

Formula for SelectKBest CHI squared test [20]

Formula

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} .$$

χ^2 = chi squared
O_i = observed value
E_i = expected value

As the Train-test split we can't just put a fixed number of features to work with in the whole project so the feature selection although it's the same the number of features differ from dataset to another (there's no magical number to feature selection)

Finalization:

For out machine learning part now it's just about building the final model. We build the same 3 model using the parameters we got from grid search and trained them on

the data after feature selection. And we choose the best model with best accuracy and classification report as a whole. For different dataset we built different model and saved the model to be working with it on API.

Issue we faced:

1.there's shortage of datasets to work with.
2. we have faced a lot of overfit and underfit with the data because either it has small number of sample or there's imbalance in the classes.

4.2 Building API and upload to live server

Fast API: [29]
a modern, fast (high-performance) web framework for building APIs with Python 3.6+based on standard Python type hints.
Heroku: [30]
Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.
GitHub: [31]
GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.
Method:
First, we build our API using Fast API and tested it on local host
Second, we uploaded the API codes to GitHub therefor we can work with it updates or modifications later with ease. We created API with 5 end point for each Cancer detection type.

The final part to link the API with Heroku server and deploy it.
Issues:
1. There're few free servers to work with.
2. linking server with Heroku wasn't optimal since Heroku had issue with GitHub.

4.3 Building Flutter Application

What Is the Flutter Framework?
Flutter is Google's free and open-source UI framework for creating native mobile applications. Released in 2017, Flutter allows developers to build mobile applications with a single codebase and programming language. This capability makes building both iOS and Android apps simpler and faster. And you can also build desktop app and wed.
Packages used in application: [32]

1. Flutter bloc
2. hex color
3. file picker
4. http
5. window manager

Application has 2 screens: [33]

- first screen to choose the type of cancer that want to detect the data
- the second screen is used to get file from user and display the result

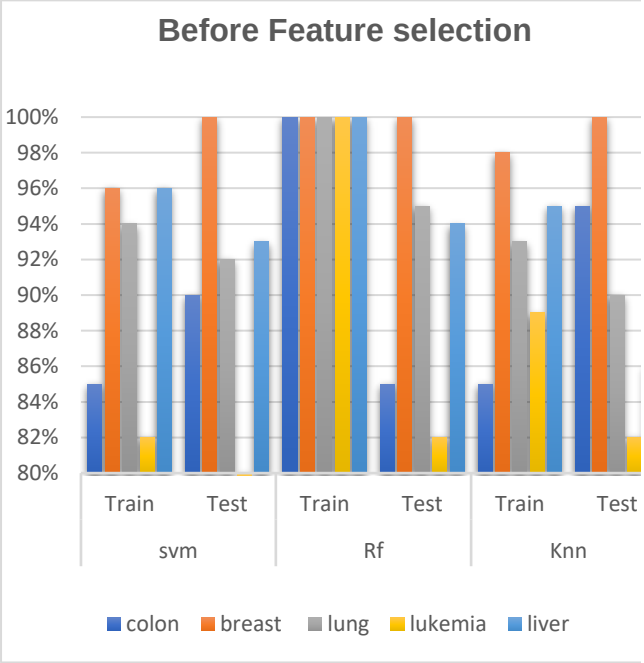
Usage of Package:
1.to have more control in application we used **flutter bloc package** it helps us to rebuild specific widget in our screen not rebuild all widgets for one change in widget, help us to write solid code also and the when the error happen.
2.We used hex color package to pick color from UI color that created in Figma app. [34]
3.We used file picker package to pick file from user device and store it.
4.After picking the file we want to send it to our API to get results, so we used HTTP package to link our application with API server to send request and get response
5.To determine application width and height we used window manager package

4.4 Result Analysis

Table1

Before Feature selection

classi-fier	type	colon	breast	lung	luke-mia	liver
svm	Train	85%	96%	94%	82%	96%
	Test	90%	100%	92%	76%	93%
Rf	Train	100%	100%	100%	100%	100%
	Test	85%	100%	95%	82%	94%
Knn	Train	85%	98%	93%	89%	95%
	Test	95%	100%	90%	82%	86%



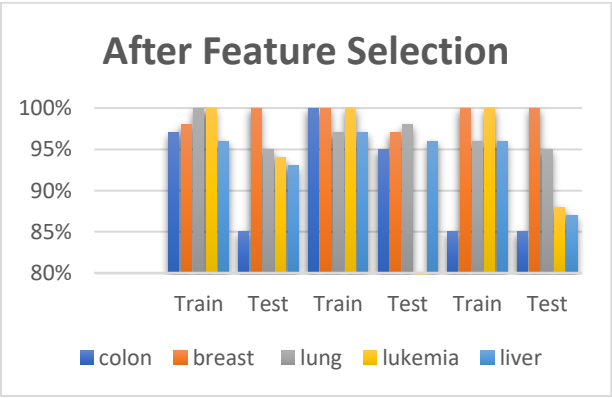
As we can see from table 1 some of the datasets suffer from overfitting and some from underfitting. We can't use these models to predict actual real-life data.
After using the feature selection and hyperparameters for building new model with only selected features from the dataset we managed to overcome some of the problems. we only choose one model for each data, the model that perform the best. Shown in the table below

Table3

type	Best model	Accuracy
Colon	random forest	95%
	GA+SVM	93%
lung	Knn	95%
leukemia	SVC	94%
liver	random forest	96%
breast	random forest	97%

Table2

		After Feature selection				
classifier	type	colon	breast	lung	lukemia	liver
svm	Train	97%	98%	100%	100%	96%
	Test	85%	100%	95%	94%	93%
Rf	Train	100%	100%	97%	100%	97%
	Test	95%	97%	98%	76%	96%
Knn	Train	85%	100%	96%	100%	96%
	Test	85%	100%	95%	88%	87%



5 Conclusion

In this examination, we assessed the utilization of machine learning techniques for the order of classification of colon cancer prediction/prognosis dependent on the variation in gene expression. We additionally examined to discover the dependability of the most significant gene expression or patterns from a natural point of view. For this reason, we have presented the results of our experiments with and without feature selection algorithm [1]. From the analysis of experimental results, we may infer that the combination of different types of features selection methods and classification models can give good outcomes in the field of detecting and classifying several categories of cancer. In future we will extend our research that can integrate more sophisticated methods for feature selection.

References

[1].
https://scikit-learn.org/stable/modules/feature_selection
[2].
https://www.pnas.org/doi/full/10.1073/pnas.96.12.6745
(Alon, U. and Barkai, N. and Notterman, D.A. and Gish, K. and Ybarra, S. and Mack, D. and Levine, A.J.)

[3]. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV
[4] Park CH, Kim SB (2015) Sequential random k-nearest neighbor feature selection for high-dimensional data. Expert Syst Appl 42(5):2336–2342
[5] Alladi SM, Shinde SP, Ravi V, Murthy US (2008) Colon cancer prediction with genetic profles using intelligent techniques. Bioinformation 3(3):130–133
[6] Ludwig SA, Picek S, Jakobovic D (2018) Classifcation of cancer data: analyzing gene expression data using a fuzzy decision tree algorithm. Springer, Berlin, pp 327–347
[7] Nguyen T, Khosravi A, Creighton D, Nahavandi S (2015) A novel aggregate gene selection method for microarray data classifcation. Pattern Recognit Lett 60–61:16–23
[8] Gao L, Ye M, Wu C (2017) Cancer classifcation based on support vector machine optimized by particle swarm optimization and artifcial bee colony. Molecules 22(12):2086
[9] Xi M, Sun J, Liu L, Fan F, Wu X (2016) Cancer feature selection and classifcation using a binary quantum-behaved particle swarm optimization and support vector machine. Comput Math Methods Med 2016:1–9
[10] Salem H, Attiya G, El-Fishawy N (2017) Classifcation of human cancer diseases by gene expression profles. Appl Soft Comput 50:124–134
[11] Aalaei S, Shahraki H, Rowhanimanesh A, Eslami S (2016) Feature selection using genetic algorithm for breast cancer diagnosis: an experiment on three diferent datasets. Iran J Basic Med Sci 19(5):476–482
[12] Chang LY, Wang HW (2006) Analysis of trafc injury severity: an application of non-parametric classifcation tree techniques. Accid Anal Prev 38(5):1019–1027
[13] Harb R, Yan XD, Radwan E, Su XG (2009) Exploring precrash maneuvers using classifcation trees and random forests. Accid Anal Prev 41:98–107
[14]. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0249094
[15]. https://sbcb.inf.ufrgs.br/cumida
[16]. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE71935
[17]. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE19804
[18]. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE42568
[19]. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE76427
[20] Cochran, William G. (1952). "The Chi-square Test of Goodness of Fit". The Annals of Mathematical Statistics. 23 (3): 315–345.
[21] Stone M. Cross-validatory choice and assessment of statistical predictions. J. Royal Stat. Soc., 36(2), 111–147, 1974.
[22] Adenso-Diaz, B., Laguna, M.: Fine-Tuning of Algorithms Using Fractional Experimental Design and Local Search. Operations Research 54(1), 99–114 (2006)
[23]. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier
[24]. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[25] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier
[26] https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
[27] https://scikit-learn.org/stable/modules/svm
[28] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[29] https://fastapi.tiangolo.com (This project is licensed under the terms of the MIT license.)

[30] https://blog.heroku.com/data-in-functions
[31] https://github.com
[32] https://pub.dev
[33] https://github.com/momaher74/CancerDetecion
[34] https://www.figma.com/file/ajm93OirtpqboJwOFhcu2o/graduate_project_uiux

Appendix
UI Screen shots