Microarray Data Analysis

(cancer detection)

Team Member

- **Mohamed Essam**
- **Mohamed Maher**
- **Mohamed Fekry**
- **Mohamed Magdi**
- **Malek Ashraf**
- **Samaa Rabea**

Introduction

- Microarray data is an 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

- 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.

- 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

Objectives

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

2. This project aims to analyze and employing a machine learning approach and feature selection technique (Univariate feature selection) .

3. 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.

Problem Definition

- Detection of colon cancer based on microarray dataset using machine learning as a feature selection and classification techniques

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

Datasets

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

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

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

Related Works

- There's a few works done on this type of cancer detection (Colon Cancer)
- They have used a lot of different feature selection method in order to get the best result
- Feature selection used:
  - PSO (Particle Swarm Optimization)
  - Genetic algorithm
  - mRMR (Minimum redundancy maximum relevance)
  - MDA (Mean Decrease Accuracy)
  - MDG (Mean Decrease Gini)
  - ABC (Artificial Bee Colony algorithm)
- Some also used a mix between 2 or 3 feature selection

Result of some related work

- Using PSO and GA with SVM classifier they have achieved accuracy of 91.90%
- Using PSO and mRMR with SVM classifier they have achieved accuracy of 90.32%
- Using PSO and GA with Naïve Bayes classifier they have achieved accuracy of 85.50%
- Using PSO and GA with DT classifier they have achieved accuracy of 83.9%

Project Roadmap

1. Building Machine learning model
    1. Data processing
    2. K-fold with GridSearch for Tuning our models
    3. Split the data with Train-Test Split method
    4. Feature selection
    5. Building the Final models
2. Building API
    1. Build API with FastApi
    2. Deploy the API with Heroku
3. Building desktop application
    1. Using Flutter and Visual Studio Tools

Data preprocessing

- Read the CSV File with pandas
- Check and remove duplicates samples
- Check and drop any Null values
- Split features from labels
- Use OrdinalEncoder with labels to change it to numerical values

Tuning the models and split data

- Define K-fold with tenfold and shuffle True
- Perform GridSearchCV on every model using CV we defined and accuracy as Strategy to evaluate (Scoring)
- Split the data with Train Test Split
  - Test_size vary from data to another

Used Models

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:

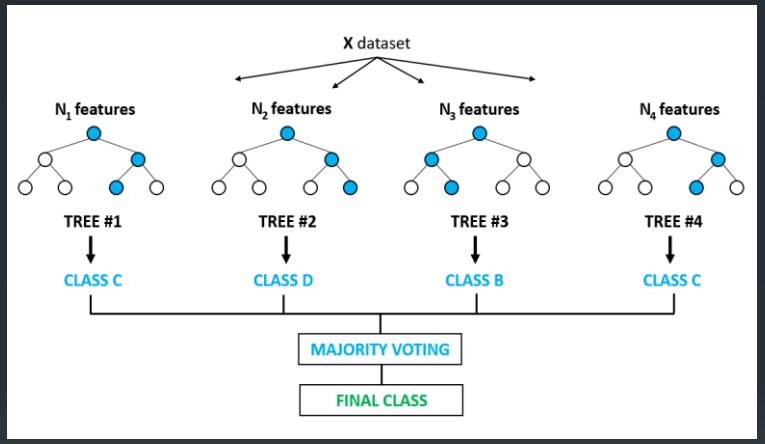
1. Random Forest Classifier
2. Support Vector Classifier (Linear)
3. K nearest Neighbor

Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning

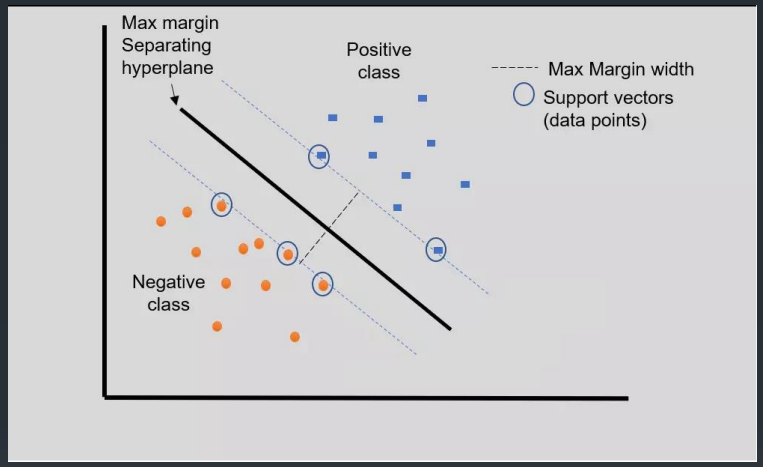
Usage :

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.



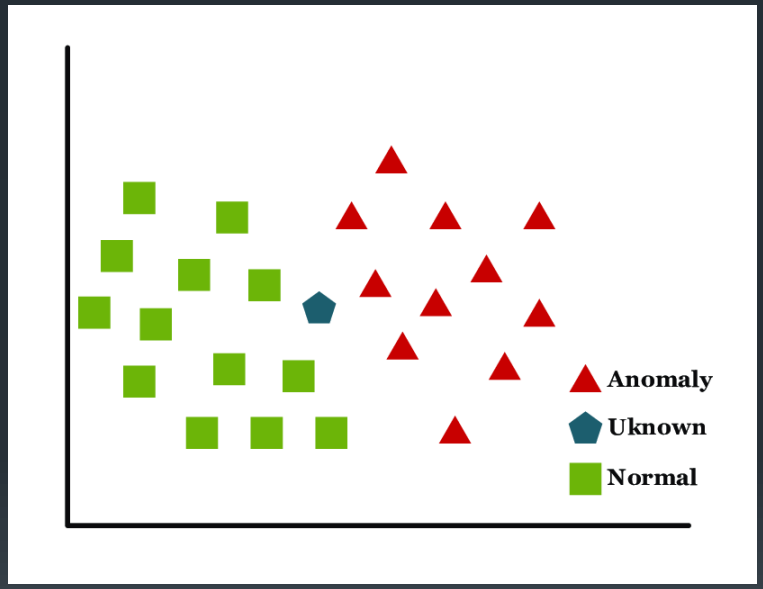
Support Vector Classifier

- A support vector machine is a supervised learning algorithm that sorts data into two categories. It is trained with a series of data already classified into two categories, building the model as it is initially trained. The task of an SVM algorithm is to determine which category a new data point belongs in. This makes SVM a kind of non-binary linear classifier.
- Usage: Biological sciences, including protein classification



K nearest Neighbor Classifier

k-NN is a widely used pattern classification technique because of its simplicity and efficiency's K-NN estimates class attribute depending the k nearest training examples in the feature space. When a dataset is given, it chooses the k nearest samples from the classified training data and determines the class considering the most representative samples. Euclidean distance similarity metric is used to select the neighborhoods



Feature selection

- We used Univariate feature selection as the backbone of our Machine learning part of project
- The number of features differ between dataset
  - There's just no magic number to work with every time and get the best result possible
- The number of features shown in the table

Type	colon	breast	lung	lukemia	liver
Number of features selected	200	5000	2000	15000	10000

Feature selection technique

- Feature selection we use is Univariate feature selection .
- Univariate feature selection 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 CHI2 squared test

Formula

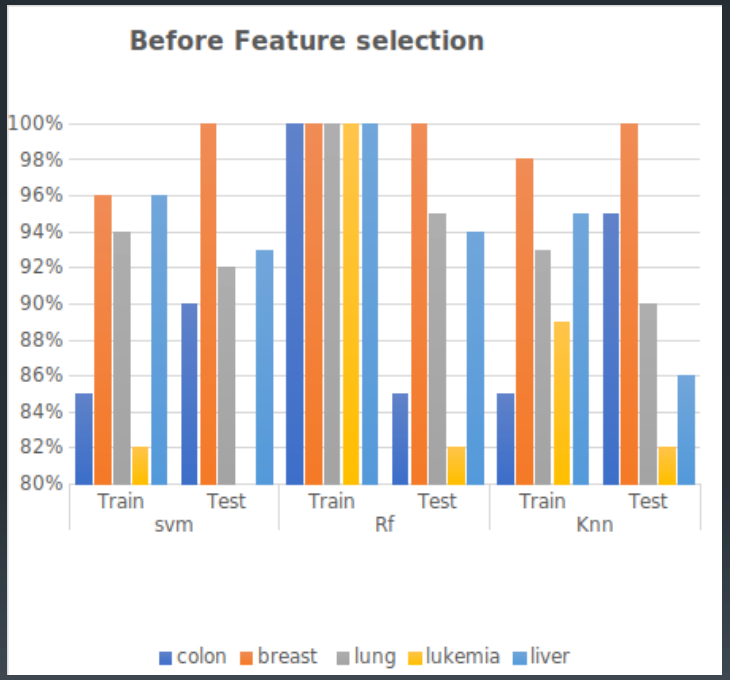
$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

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

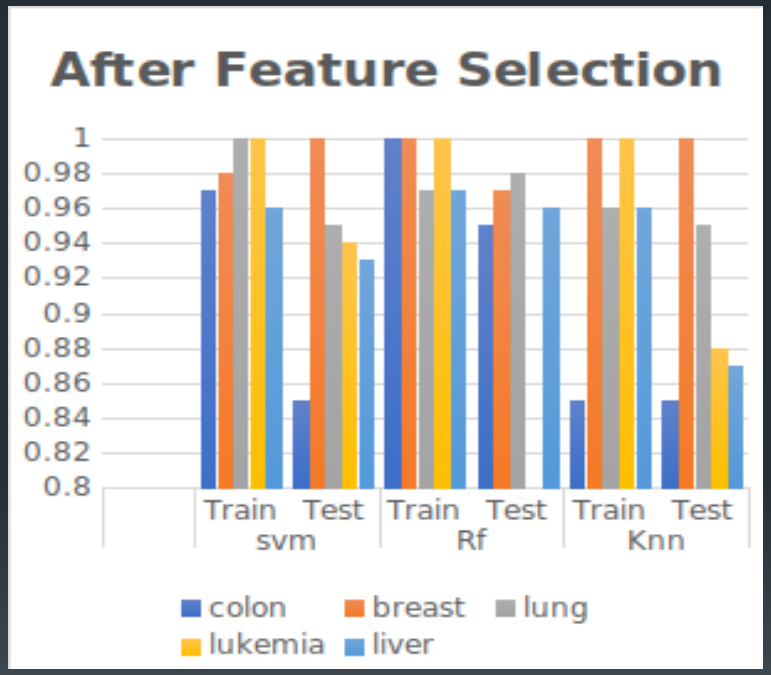
Result Analysis

Before Feature selection						
classifier	type	colon	breast	lung	leukemia	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%



After Feature selection

After Feature selection						
classifier	type	colon	breast	lung	leukemia	liver
svm	Train	97%	98%	100%	100%	96%
	Test	85%	100%	95%	94%	93%
Rf	Train	100%	100%	97%	100%	97%
	Test	95%	97.5%	98%	76%	96%
Knn	Train	85%	100%	96%	100%	96%
	Test	85%	100%	95%	88%	87%

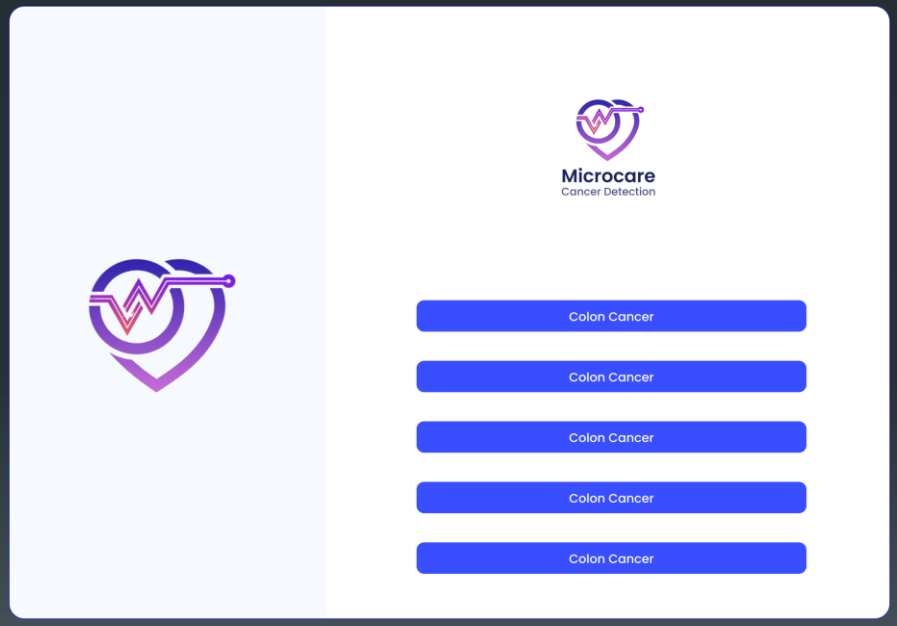


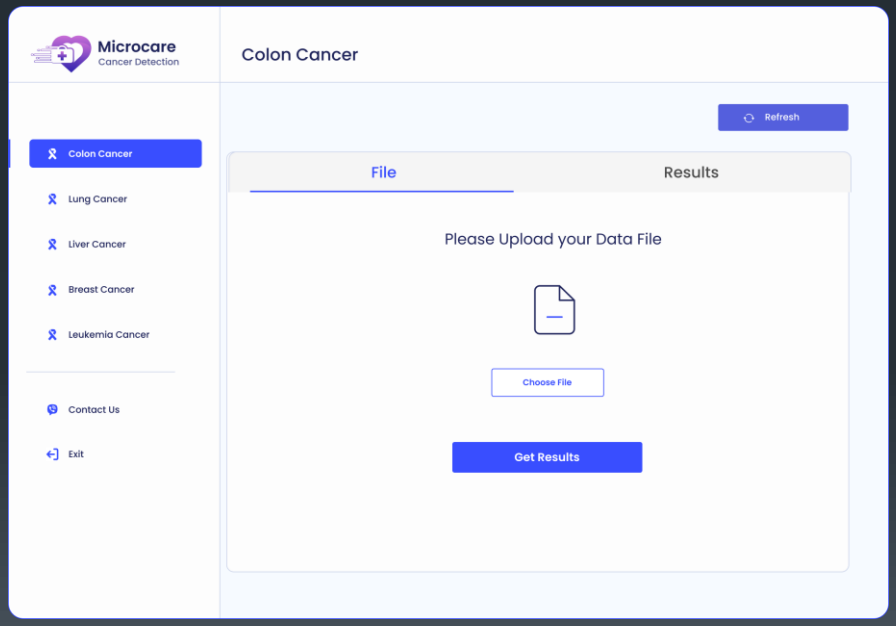
Final models

After using the feature selection and hyperparameters for building new model with only selected features from the dataset we managed to overcome the problems like you can see in table 6, but we only choose one model for each data, the model that perform the best, no overfit or underfit. the models we choose are shown in the table7 below

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%

Application UI





Future work

- **Improve the App UI**
- **Add feature selection as option to ease the app usage**
- **Look for more data to work with**