MINISTRY OF HIGHER EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY JSC

FACULTY OF COMPUTER TECHNOLOGY AND CYBERSECURITY

Ibrayeva M.A
Ibrayeva T.D.

Early Detection of Breast Cancer Using Machine Learning Techniques

DIPLOMA PROJECT

Major 6B06112 – Data science

Almaty 2023

MINISTRY OF HIGHER EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN
INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY JSC

DEPARTMENT OF MATHEMATICAL AND COMPUTER MODELLING

Approved

Head of Department,

PhD, assistant professor

_____Ydyrys A.

«____»_____2023

DIPLOMA PAPER

Early Detection of Breast Cancer Using Machine Learning Techniques

Major 6B06112 – Data science

Done by: Ibrayeva M.A. _____
«____»_____2023 (signature)
Ibrayeva T.D. _____
«____»_____2023 (signature)

Research advisor: PhD,
Associate Professor
Nurtas M. _____
«____»_____2023 (signature)

Reviewer: PhD,
Professor of KBTU
Issakhov A.A. _____
«____»_____2023 (signature)

Almaty 2023

International Information Technology University

Faculty of Computer Technology and Cybersecurity

Department of Mathematical and Computer Modeling

Major – 6B06112 – Data Science

Diploma Project Assignment

Students

Ibrayeva M.A, Ibrayeva T.D.

Diploma work (project) topic

Early Detection of Breast Cancer Using Machine Learning Techniques.

Approved by IITU order № ____ dated «____»_____20__

Diploma work (project) submission date «____»_____20

Diploma work (project) initial data

Breast Cancer Wisconsin (Diagnostic) Dataset

Details of computations and explanations (list of issues due to be addressed)

1. Analysis of existing breast cancer detection methods
2. Data preprocessing and feature extraction
3. Deep learning methods
4. Extracting of meaningful features from dataset
5. Training and evaluation of different models
6. Validation and performance assessment
7. Documentation and reporting

CD containing the digital version of diploma paper and attachments

Provided

Consultations on diploma work (project) (with related project chapters named)

Consultant	Name	Signature, date	
		Assignment given	Assignment received
English language consultant	Senior-lecturer Abdulina M.		
Compliance monitor	Assistant professor Abdikalikova Z.T.		

Date «___»_____20_

Research advisor _____

(signature)

Assignment received by _____

(signature)

Diploma project writing schedule

Ibrayeva M.A, Ibrayeva T.D.

Title: **Early Detection of Breast Cancer Using Machine Learning Techniques.**

№	Assignment	Submission date
1	Creation of the graduation paper writing Schedule	December 1
2	Searching dataset, processing and analyzing Information	December-January
3	Model training and evaluation	December-April
4	First pre-defense	February 17
5	Drafting and submission to the Research advisor (Introduction, Chapter 1)	March
6	Drafting and submission to the Research advisor (Chapter 2, Conclusion)	April
7	Revision of the diploma paper with due consideration advisor's comments	April 21-22
8	Second pre-defense	April 27-28
9	Submission of the diploma paper to the English language consultant	May 2 – 10
10	Submission of the diploma paper to the plagiarism check-up	May 2 –28
11	Submission of the diploma paper to the compliance monitor	May 2 – June 2
12	Submission to the reviewer for approval	May 2 – 30
13	Diploma work (project) defense	June 5-9

Student: Ibrayeva M.A. _____

(signature)

Student: Ibrayeva T.D. _____

(signature)

Research advisor: Nurtas M. _____

(signature)

Date «___»_____20___

АҢДАТПА

Сүт безі қатерлі ісігі әлемдегі, әсіресе әйелдер арасында кең таралған қатерлі ісік түрлерінің бірі болып табылады. Қатерлі ісік ауруын ерте анықтау тиімді емдеу және пациенттің өмір сүруі үшін өте маңызды.

Дипломдық жұмыстың мақсаты сүт безі обырын ерте анықтау үшін машиналық оқытуға негізделген жүйені құру. Ұсынылған жүйемаммографиялық кескіндерді қатерсіз және қатерлі деп жіктеу үшін бірнеше машиналық оқыту әдістерін пайдаланады, сонымен қатар 569 пациент және 10 нақты бағаланған санат туралы деректері бар ашық кітапхананы пайдаланды.

Дипломдық жобада 46 бет, 33 иллюстрация, 31 пайдаланылған әдебиеттер бар.

Кілт сөздер: ДЕРЕКТЕР ТУРАЛЫ ҒЫЛЫМ, АНАЛИЗ, ДИАГНОСТИКАЛЫҚ МЕТОДТАР, МЕДИЦИНАЛЫҚ ЗЕРТТЕУЛЕР, МАШИНДЫ ОҚЫТУ, МАММОГРАММА, КЛАССИФИКАЦИЯ, НЕЙРОНТЫҚ ЖЕЛІЛЕР, МОДЕЛДІ ОҚЫТУ, КЕЛЕНІ ЖОГАРЫ ОҚЫТУ.

АННОТАЦИЯ

Рак молочной железы является одним из самых распространенных видов рака в мире, особенно среди женщин. Раннее выявление рака имеет решающее значение для эффективного лечения и выживания пациентов.

Целью дипломной работы является создание системы на основе машинного обучения для раннего выявления рака молочной железы.

Предлагаемая система использует несколько методов машинного обучения для классификации маммографических изображений на доброкачественные и злокачественные, а также использует открытую библиотеку с данными о 569 пациентах и 10 реально значными характеристиками.

Дипломный проект содержит 46 страниц, 33 иллюстраций, 31 ссылок.

Перечень ключевых слов: НАУКА О ДАННЫХ, АНАЛИЗ, ДИАГНОСТИЧЕСКИЕ МЕТОДЫ, МЕДИЦИНСКИЕ ИССЛЕДОВАНИЯ, МАШИННОЕ ОБУЧЕНИЕ, МАММОГРАММА, КЛАССИФИКАЦИЯ, НЕЙРОННЫЕ СЕТИ, ОБУЧЕНИЕ МОДЕЛЕЙ, ГЛУБОКОЕ ОБУЧЕНИЕ.

ABSTRACT

Breast cancer is one of the most common types of cancer in the world, especially among women. Early detection of cancer is critical for effective treatment and patient survival.

The purpose of the diploma project is to create a system based on machine learning for the early detection of breast cancer. The proposed system uses several machine learning methods to classify mammography images into benign and malignant, and also used an open library with data on 569 patients and 10 real-valued features.

The explanatory note consists of three chapters, which set out the theoretical and practical foundations of the concept of the system.

The diploma project contains 46 pages, 33 illustrations, 31 references.

A keywords list: DATA SCIENCE, ANALYSIS, DIAGNOSTIC METHODS, MEDICAL RESEARCH, MACHINE LEARNING, MAMMOGRAM, CLASSIFICATION, NEURAL NETWORKS, MODEL TRAINING, DEEP LEARNING.

CONTENTS

LIST OF TERMS AND ABBREVIATIONS

1. **SVM -** Support Vector Machines

A type of supervised learning algorithm that can be used to classify mammograms into twoclasses (normal or malignant).

2. **CAD –** Computer-Aided Detection

CAD is a technology that uses machine learning algorithms and image processing techniques to analyze medical images, such as mammograms, and identify potential areas ofconcern that may be indicative of disease.

3. **ANN –** Artificial Neural Network

Artificial Neural Networks is a type of machine learning algorithm, can be used for both classification and regression tasks, including breast cancer detection at early stages. In ANN, a network of artificial neurons is created to learn and recognize patterns in the inputdata.

4. **API –** Application Programming Interface

An API, or Application Programming Interface, is a set of protocols, routines, and tools forbuilding software applications. It provides a standardized way for different software systems to communicate and exchange data with each other.

INTRODUCTION

The relevance of our project is the development of new system to detect breast cancer. This disease is a leading cause of cancer-related deaths in women worldwide. Early detection and treatment of breast cancer significantly improve patient survival rates. Mammography is the most widely used screening method for breast cancer. However, the interpretation of mammogram images can be challenging, and radiologists may miss small or subtle abnormalities that could be indicative of early-stage breast cancer. Therefore, there is a growing interest in developing computer-aided diagnostic systems to assist radiologists in the detection of breast cancer.

Methods that we used is machine learning techniques, such as logistic regression, random forest, and support vector machines (SVM), have shown great promise in the field of medical imaging for the automated detection of breast cancer. These techniques can analyze mammogram images and classify them into benign or malignant cases with high accuracy, sensitivity, and specificity. Such automated systems have the potential toreduce human error and variability in interpretation, as well as improve the efficiency and accuracy of breast cancer screening. Several studies have shown promising results, with machine learning algorithms achieving high accuracy rates in detecting early-stage breast cancer. For example, a recent study published in the Journal of the National Cancer Institute found that a machine learning algorithm was able to identify breast cancer in mammograms with an accuracy of 90%, which is comparable to that of trained radiologists. [1]

Moreover, the development of deep learning algorithms has shown even greater potential in detecting early-stage breast cancer. Deep learning algorithms can automatically learn and identify patterns in data and have been shown to achieve high levels of accuracy in detecting breast cancer in mammograms.

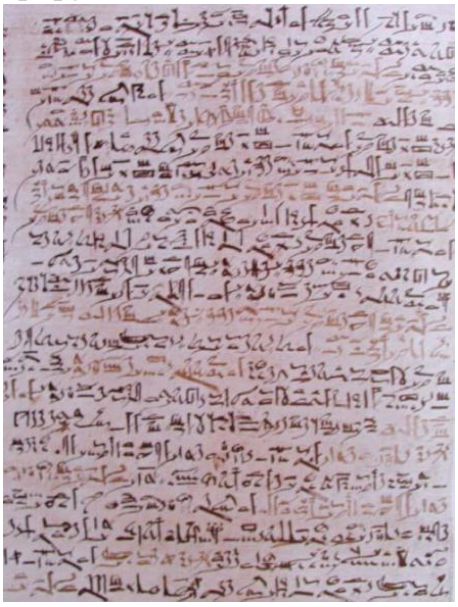
This diploma project aims to develop a machine learning-based system for the early detection of breast cancer. The proposed system will be trained on a large dataset of mammogram images and evaluated using various performance metrics. The goal is to create a reliable and efficient tool for assisting radiologists in the early detection of breast cancer and improving patient outcomes. The study will contribute to the growing body of research on the use of machine learning techniques in medical imaging and could have significant implications for diagnosis and treatment of breast cancer.

1 THEORETICAL BACKGROUND OF THE PROBLEM

1.1 History

The previous events and artifacts that have been preserved tell the story of how people have battled breast cancer throughout history. It is an epic voyage that chronicles the evolution of illness from the belief in malicious spirits or angry deities to the understanding of observable physical causes and the advancement of therapeutic arts, from mysticism to the latest technologies of modern science.

The first records of breast cancer date back to Babylonian times. However, the earliest information regarding breast cancer is contained in the Imhotep papyrus, which was kept in the tomb until 1862 and then sold to Edwin Smith (Picture 1.1.1). Physicians were already aware that some breast cancers were difficult to cure before the third millennium BC. The translated Surgical Papyrus of Edwin Smith, one of the eight remaining Egyptian medical papyri, contains the first reference to breast cancer.



Picture 1.1 – Illustration showing Edwin Smith Surgical Papyrus, a copy of the first document believed to describe cancer of the breast, circa 3000 BC

Hippocrates described breast cancer instances in great detail. He recounted the example of a woman from Abdera who had breast cancer and crimson flow from the nipple in one of his medical histories. Hippocrates observed that bleeding had a positive effect but also that the woman died after the bleeding stopped. Additionally, he attempted to revive menstruation in young patients and connected the cessation of menstruation to breast cancer. Hippocrates' in-depth account of the developmentof breast cancer is still believable today. [2]

He found that solid tumors, which do not contain pus and spread to other body areas, begin to form in the mammary gland and subsequently solidify. The patient's symptoms worsen as the illness progresses, including irritability, refusal to eat, shooting pains from the chest to the neck and shoulder blades, complaints of thirst, and exhaustion. Death was going to happen at that point. Because it was ineffective and shorted the patient's life, he advised against treating latent breast cancer.

Breast cancer research took a step forward with the work of Roman physician Aulus Cornelius Celsus in the 1930s. Celsus recognized that breast cancer was prevalent among women and detailed four stages of the disease in his book "De Medicina": cacoethes, carcinoma without skin ulcers, carcinoma with ulcers, and thymus. While Celsus recommended surgical removal of the cacoethes stage, he advised against treating other stages. When unsure, severe treatment was first applied to the tumor and if symptoms improved, it was classified as cacoethes. Conversely, if symptoms worsened, it was classified as cancer. Neoplasms such as fibroadenomas, phyllodes tumors, and even tuberculosis, responded well to treatment. During that time, the first stage was effectively treated, and researchers began experimenting with subsequent stages. [2]

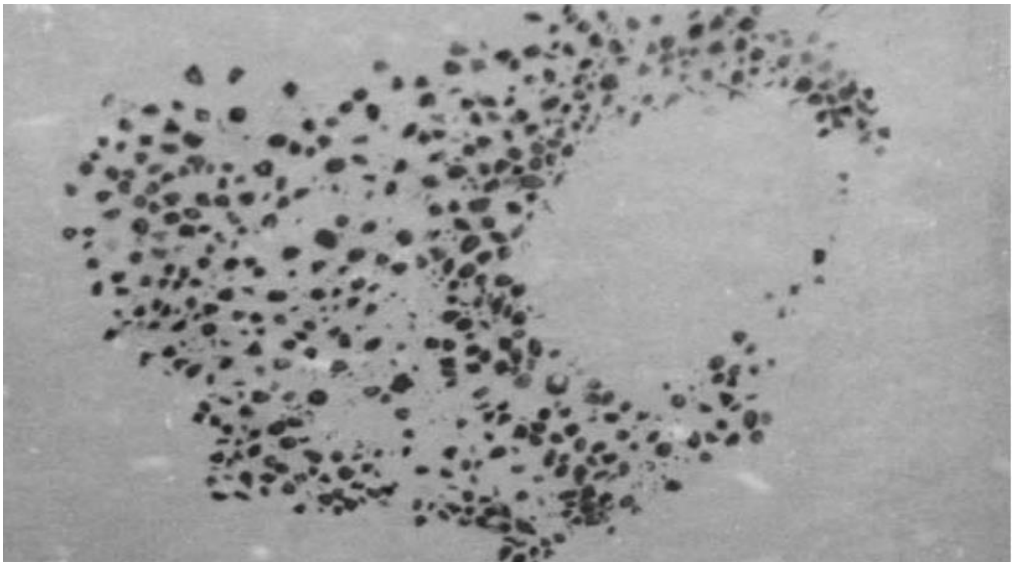
Breast tumors have been investigated by scientists globally, who have put forward various treatment methods. Despite Galen's less modern approach, medicine ceased to progress after him. In contrast, Leonid's work highlighted that breast cancer can spread to the axillary cavity, and he didn't specifically endorse surgical intervention, but rather selective excision. Galen's treatment method involved using "folk" remedies, cleansing, diet, and local treatments, with other methods only being used if the cancer progressed. [2]

During the late Middle Ages, Henri de Mondeville, who served as the surgeon to the King of France between 1260-1320 AD, improved upon Galen's concept of black bile. He provided a more detailed description of true breast cancer, characterizing it as an ulceration with thick edges and an unpleasant odor. De Mondeville recognized that when surgical removal is not complete, the wound maynot heal, leading to a higher likelihood of disease recurrence.

Without anesthesia or antiseptics, cancer treatment was a painful and dangerous trial, usually conducted in the patient's home. According to Richard Wiseman's (1622-1676) report, who was the surgeon to Charles II, out of the twelvesurgeries performed, two patients died due to the procedure, eight died soon after due to advanced cancer, and two out of the 12 were considered "cured" for an unspecified duration. [2]

With tremendous advancements in human pathology and surgical safety, the eighteenth century was undoubtedly a giant stride forward for oncology. Ignaz Semmelweis (1818–1865), a physician from Hungary, and Oliver Wendell Holmes, MD (1809–1894), a Harvard University professor of anatomy and physiology, both advocated for hand washing. By utilizing carbolic acid spray, Joseph Lister (1827-1912) launched surgical antiseptics in Glasgow in 1867, building on Louis Pasteur's (1822–1955) discovery of "rotting" germs. [3] Further reducing pollution was the development of aseptic techniques, such as the first use of steam sterilization in 1886 by Berlin-based Ernst von Bergmann, the invention of surgical masks in 1886 by the Pole Johannes von Mikulicz-Radecki, and the use of sterile rubber surgical gloves in 1890 by William S. Halsted. [4]

Anesthesia allowed surgery to go quickly and allowed medical professionals to concentrate on the process itself rather than the hurry of the procedure. The development of breast treatments benefited greatly from the use of the microscope as well. It assisted Johannes Muller (1801–1859) in his discovery that cancer is made out of live cells. He may have been the first to explain how cancer spreads locally and how cancerous cells divide according to the principle of metastasis. [5]



Picture 1.2 – Illustration showing the cellular structure of breast cancer

Mastectomy was seen as a less successful alternative than other therapies for breast cancers during the end of the nineteenth century, but it was still preferred to no therapy. Five years following the start of the first symptom, the actuarial survival rate of the first fifty Halsted cases was 40.4%, more than double the 18%[6][7] rate for untreated patients referred to the charity department of Middlesex Hospital in London between 1805 and 1933. [4]

During this time, two occurrences were critical for the future management of breast cancer. The first was the invention of X-rays, and the second was the understanding that hormones play a role in breast cancer.

Radiation treatment and mammography were made possible by Wilhelm Conrad Roentgen's discovery of X-rays in Würzburg in 1895. The enigmatic laser, bearing the letter "x," not only entered the body but also destroyed cancer cells. As a result, mastectomy was subsequently followed by radiation therapy as a postoperative (and occasionally prior) addition, which led to the development of breast-preserving surgical surgery. [2]

Ovariectomy—the removal of the ovaries—was the first step in the hormonal therapy of breast cancer. Theoretically, this strategy slowed the progression of the disease, but this was not totally true, even if it encouraged pharmaceutical research to discover a cure.

The introduction of mammography and chemotherapy, as well as the rejection of aggressive surgery, occurred throughout the following 100 years. Parallel to surgery, mammography emerged as the most significant advancement inbreast cancer detection to date. Early medical professionals understood that tiny breast cancer was more treatable. Numerous forms of clinically latent breast cancers, including the frequently curable ductal carcinoma in situ, might be found thanks to mammography.

After mammography, a variety of cutting-edge breast imaging devices were developed. The dry process approach of Xeromammography did not persist very long. [8] It had excellent detail records of every breast cancer structure and could be inspected without viewing windows. However, with advancements in mammography using a film screen, it was no longer in use. Mammography was expanded to include ultrasound tests and magnetic resonance imaging (MRI). In the 1950s, ultrasonography first came into usage. It might discriminate between cysts and solid forms, describe solid formations, and enable a real-time, needle- controlled biopsy of suspected lesions without irradiation. Ultrasound was not always able to reveal malignant tumors that were found by other techniques, and the findings were highly operator-dependent, making it unsuitable for population screening. MRI has shown to be useful in unique circumstances.

Monitoring the progression of the disease and its treatment was a crucial component of the research, which meant that the postoperative period was equally crucial. The overall cure rates across the various treatment groups were found to be comparable whether the regional nodes or the whole breast were removed.

When Paul Ehrlich, a researcher, first used the word "chemotherapy," it referred to a systemic therapy that frequently caused advanced types of breast cancer to temporarily retreat and occasionally completely vanish. [9]

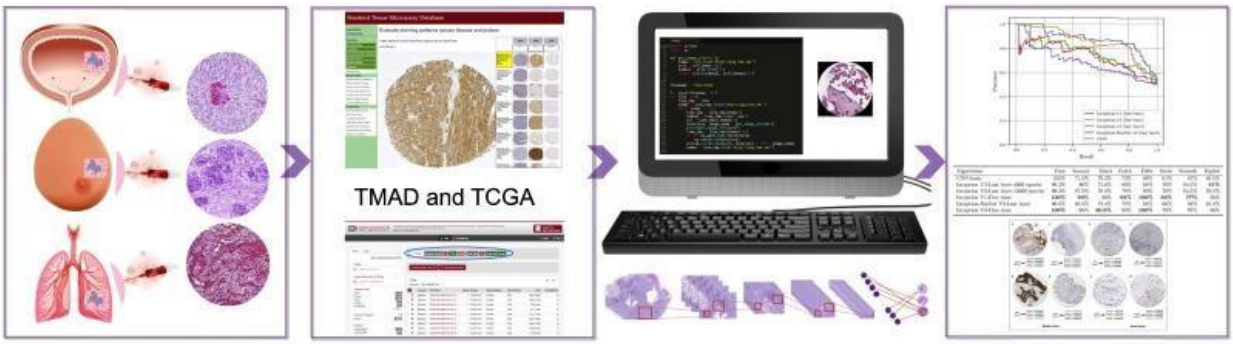
The inclusion of chemotherapy changed the way that breast cancer was treated, requiring a team of experts to work together to treat both the local and systemic aspects of the illness. These doctors employed a mix of surgery, radiationtherapy, and systemic chemo-hormonal therapy.

Breast cancer was known to be an uncontrolled cell development by the end of the 20th century, but its specific origin was still unknown. Breast cancer remains a worrisome problem despite contemporary medical developments and diagnostic techniques that have made it simpler to effectively diagnose the illness. Despite three millennia of medical progress, statistics show that there is still a probability of contracting the disease today.

Breast cancer is a significant public health issue, and early detection is crucial for effective treatment and patient survival. According to Petropavlovsk.news, the Kazakhstani Ministry of Health reports that more than 30,000 new instances of cancer are discovered each year. 36,000 cases of malignant illnesses were reported in Kazakhstan in 2021. Of these, 13,000 independently sought medical care, 18,000 had malignancies found through preventative measures, and 1,800 were found through screening. Women accountfor 57.4% of instances of oncological illnesses, compared to 42.6% for males.

Women are being diagnosed with breast cancer at a higher rate. 43,200 people were affected by the illness in 2021 as a consequence of 5,000 women receiving a diagnosis. The incidence rate of cancer in Kazakhstan was 199.2 cases per 100,000 people in 2022 (compared to 190.2 cases per 100,000 in 2021, or 36,127 cases), representing a 4.7% increase rate. In the hierarchy of oncological illnesses, breast cancer typically holds the top spot (13.2%, 5,166 cases). [10]

Since the middle of the 1980s, automated data analysis and artificial neural networks have been employed to identify and diagnose cancer. The most cutting-edge techniques used today rely on machine learning. Datasets are collections of many pictures (obtained from CT scans, MRIs, mammograms, or histology) that are used to train machine learning algorithms. The photograph is put into the system, which ranks the list of research from highest to lowest chance of pathology. As a result, the doctor starts by looking at the pictures of the patients for whom the system has indicated the possibility of a malignancy. As an alternative, the expert studies an image where the AI has indicated a pathologicalregion and adds their findings to the AI's description. [11]



Picture 1.3 - Illustration showing the pipeline, which includes extracting data, training and evaluation of ANN algorithms, and prediction of various classes

One of the first researchers to use machine learning to detect breast cancer at an early stage was Dr. Stephen T. Wong, a professor of radiology, and director of the Bioinformatics and Signal Transduction Laboratory at Houston Methodist Research Institute.

Dr. Wong and his team developed a computer-aided diagnosis system in the early 2000s that combined machine learning algorithms with medical imaging to improve the accuracy of breast cancer detection. Their system analyzed mammograms and used machine learning techniques to detect and classify potential breast lesions. [14]

According to Dr. Yun Kyung Kim of Severance Hospital and Yonsei University College of Medicine in Seoul, "In practice, AI-CAD (computer-aided detection) can replace the role of second readers in double-reading settings or reduce the workload of radiologists in sorting part of mammography." Accordingto test results, 89% of 160 breast cancer cases were found utilizing AI with an anomaly score of at least 0.1 as opposed to 76% by experts. [12]

Mammography is the most widely used screening method for breast cancer, but it has some limitations, such as high false-positive and false-negative rates.
Therefore, there is a growing interest in developing computer-aided diagnostic(CAD) systems to assist radiologists in the detection of breast cancer. [15]

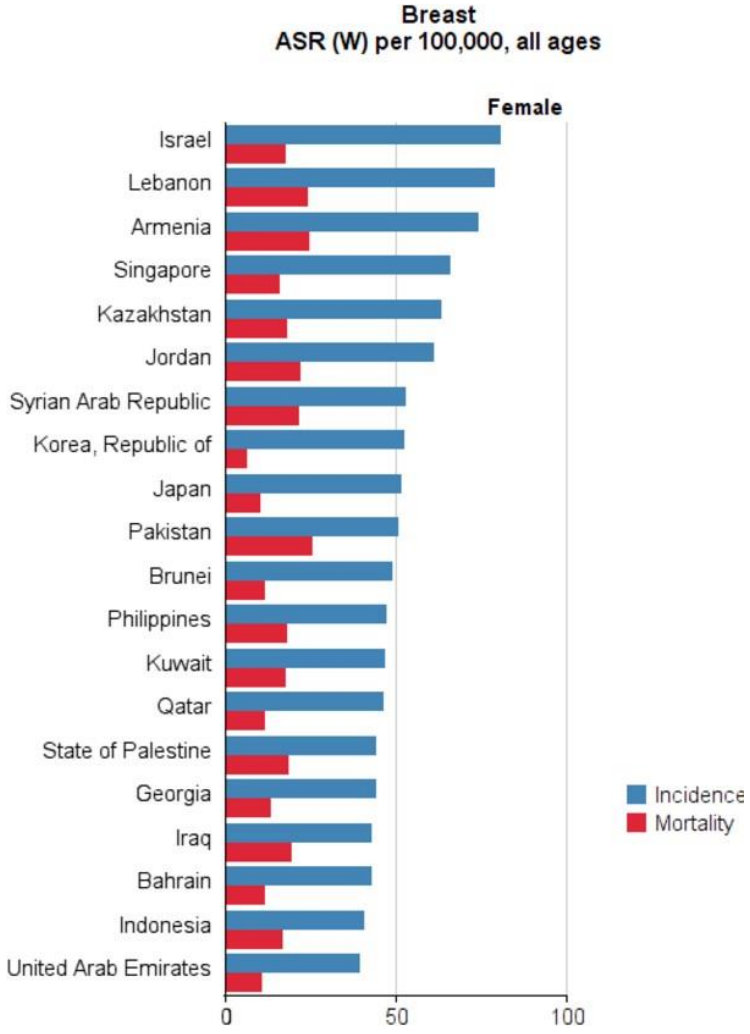
Due to thick parenchymal lesions, inadequate placement, perception difficulties, or interpretation errors, mammography may miss up to 30% of breast cancer cases. Machine learning significantly improves (15–25%) the accuracy of predicting susceptibility to cancer, recurrence, and death, despite the growing dependence on protein biomarkers in cancer diagnosis and the bias towards breastcancer detection. [11]

Machine learning techniques have shown great potential in the field of medical imaging for the automated detection of breast cancer.

1.2 Methodology and research

1.2.1 The problem statement

The purpose of our project is to create a system that will detect breast cancer on the early stages. The use of machine learning for breast cancer prediction has the potential to revolutionize the field of cancer diagnosis, providing health care professionals with a powerful tool to improve patient outcomes. By enabling earlier detection and treatment of breast cancer, machine learning can help save lives and reduce the burden of this devastating disease. In this report, we explore the use of machine learning techniques for the early detection of breast cancer. To do this we will need to train and work with data that contains useful features of patients, and with machine learning techniques we can make predictions.



Picture 1.4 – Illustration showing Incidence and Mortality of Breast Cancer and their Relationship in Asia

1.2.2 The aim of the project

The aim of our system in this project is to improve the accuracy and efficiency of breast cancer detection, as well as to increase the likelihood of successful treatment outcomes. Machine learning algorithms can be trained to analyze large datasets of mammograms and identify patterns or abnormalities thatmay be indicative of early-stage breast cancer. By identifying potential areas of concern at an early stage, healthcare providers can take proactive measures to diagnose and treat breast cancer before it progresses to more advanced stages, which can be more difficult to treat.

Overall, the goal of this project is to use technology to improve breast cancer screening and detection, ultimately leading to better patient outcomes and reduced healthcare costs.

1.2.3 Methods and research

Breast cancer is a leading cause of cancer-related deaths among women worldwide. Early detection of breast cancer is crucial for successful treatment outcomes. Machine learning algorithms have shown promise in improving the accuracy and efficiency of breast cancer detection.

One method of using machine learning for early detection of breast cancer is by analyzing mammograms. Researchers have developed various machine learning algorithms to analyze mammograms and identify patterns or abnormalities that may be indicative of early-stage breast cancer.

Another approach is to use machine learning algorithms to analyze patient data, such as genetic information, family history, and lifestyle factors to identify individuals who may be at higher risk of developing breast cancer. This approach can help healthcare providers to take proactive measures to prevent breast cancer or detect it at an early stage.

Overall, machine learning has the potential to improve breast cancer screening and detection, leading to better patient outcomes and reduced healthcare costs. However, further research and development are needed to refine and optimize these methods for clinical use.

The implementation of this project was carried out using an open-source dataset that consists of data from 569 patients and 10 real-valued features. All methods that we used in the system to make predictions described below.

Supervised Learning: This method involves training a machine learning algorithm on labeled data (data with pre-assigned classes or labels) to predict the class or label of new, unseen data. In the case of breast cancer detection, mammograms with pre-assigned labels (either normal or malignant) can be used totrain a supervised learning algorithm to detect early signs of breast cancer in new mammograms.

Unsupervised Learning: This method involves training a machine learning

algorithm on unlabeled data (data with no pre-assigned classes or labels) to identify patterns or anomalies in the data. Unsupervised learning can be used to detect potential areas of concern in mammograms that may require further examination. [16]

Logistic regression is a supervised learning algorithm that can be used for binary classification tasks, such as detecting the presence or absence of breast cancer. In logistic regression, the algorithm fits a logistic function to the input data
to estimate the probability of a particular outcome (in this case, whether the mammogram is indicative of breast cancer or not). The logistic function maps any real-valued input to a probability value between 0 and 1. Based on this probabilityvalue, the algorithm can make a prediction as to whether the mammogram indicates the presence of breast cancer or not. [17]

Decision Trees: This method involves creating a tree-like model of decisions and their possible consequences. Decision trees can be used to classify mammograms as normal or malignant based on a set of pre-defined criteria. [18]

Random forest is a supervised learning algorithm that can be used for classification tasks, including breast cancer detection. In a random forest model, multiple decision trees are constructed from randomly selected subsets of the input data. Each tree makes a prediction as to the class or label of the input data, and the final prediction is made by combining the predictions of all the individual trees.
Random forest can be more accurate than a single decision tree because it reduces overfitting and variance, while increasing stability and robustness. [19]

Support Vector Machines (SVMs): SVMs are a type of supervised learning algorithm that can be used to classify mammograms into two classes (normal or malignant). SVMs work by finding the optimal boundary between the two classes, maximizing the distance between the boundary and the nearest data points. [20]

Gaussian Naive Bayes is a probabilistic machine learning algorithm that can be used for classification tasks. It assumes that the features are independent of each other and that each feature is normally distributed. In the case of breast cancer detection, the features could be characteristics of the breast tissue, such as size, texture, and shape. [21]

K-Nearest Neighbors (KNN) is a supervised learning algorithm that can beused for classification tasks, where the goal is to predict the class of a new data point based on its similarity to existing data points in a labeled dataset. The KNNalgorithm works by calculating the distance between the new data point and eachof the existing data points in the dataset. The K nearest neighbors are then identified, and the class of the new data point is assigned based on the majority class among its K nearest neighbors. [22]

Artificial Neural Networks (ANN) are machine learning algorithms that are based on the form and function of biological neurons in the brain. ANN may be used for both classification and regression applications, including early identification of breast cancer. A network of artificial neurons is built in ANN to learn and detect

patterns in incoming data.

The ANN algorithm is trained on a huge dataset of labeled mammogram images to discover the underlying patterns and traits that are predictive of early-stage breast cancer. Each neuron in an ANN layer receives input from the previous layer and produces output for the following layer. The training procedure entails modifying the weights and biases of the network's neurons in order to minimize the error between expected and actual output. Once trained, the ANN model may be used to categorize new, unlabeled mammograms as malignant or benign.

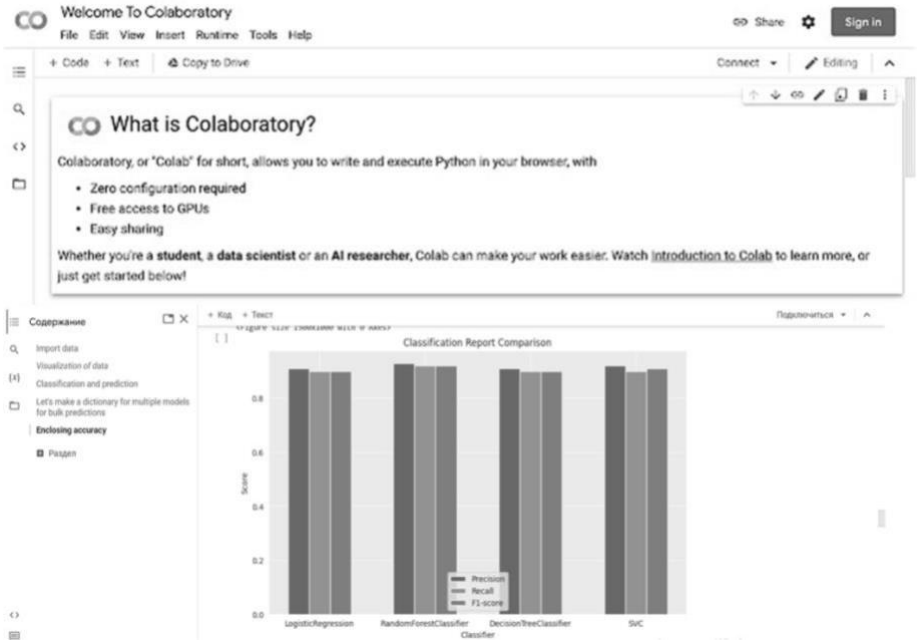
The power of ANN to learn and recognize complicated patterns and characteristics in input data is one of its many benefits for breast cancer diagnosis. ANN can also deal with noisy and missing data, and it is resistant to tiny changes in the input data. ANN has been demonstrated to attain high levels of accuracy in early-stage breast cancer diagnosis and is frequently used in conjunction with othermachine learning algorithms to boost accuracy. [23]

2 DEVELOPMENT OF SYSTEM

2.1 Choosing a platform, tools, and its description

To implement our system, we chose the **Google Colab** coding platform. Platform is a convenient and easy-to-use way to run Jupyter notebooks on the cloud.

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with images, HTML, LaTeX and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You caneasily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. [24]



Picture 2.1 - Illustration of Google Colab UI and Python Code

We use Python programming language because of its ease of use, flexibility, and vast collection of libraries and tools designed specifically for data science and machine learning. Python is a high-level language that is easy to read and write. It has a simple syntax that makes it easier to learn and understand, especially for beginners. Also used libraries such as:

Scikit-learn is a popular machine learning library that provides various algorithms for classification, regression, clustering, and more. It also provides toolsfor model selection, preprocessing, and evaluation. Scikit-learn includes implementations of the Gaussian Naive Bayes algorithm, as well as other classification algorithms that can be used for breast cancer detection. [25]

NumPy is a fundamental library for scientific computing in Python. It provides support for array and matrix operations, which are commonly used in machine learning algorithms. NumPy can be used to preprocess and manipulatedatasets for breast cancer detection. [26]

Pandas is a library for data manipulation and analysis. It provides tools for reading and writing data in various formats, as well as functions for handling missing data and performing operations on datasets. Pandas can be used to load and preprocess datasets for breast cancer detection. [27]

Matplotlib is a plotting library for creating visualizations in Python. It provides functions for creating various types of plots, including scatter plots, histograms, and line plots. Matplotlib can be used to visualize the results of breastcancer detection models. [28]

Pickle library in Python is a powerful tool for saving and loading Python objects in a serialized form. It allows to save trained machine learning models, so that they can be used later without the need for re-training the model every time. This can save time and computational resources, especially in scenarios where thetraining dataset is large and the training process is time-consuming. [29]

Seaborn library in Python is a popular data visualization library built on top of the matplotlib library. It provides a high-level interface for creating beautiful and informative statistical graphics. This library can be used to visualize the relationships between different features of the dataset, and to identify patterns thatmay be useful in predicting the likelihood of breast cancer. [30]

Keras is a high-level neural networks API in Python that was developed to make it easier to build and experiment with deep learning models. It allows us to build deep learning models that can classify breast cancer patients based on various features such as age, tumor size, and cell shape. These models can take in the features as input and predict the likelihood of a patient having breast cancer as output. [31]

```
[ ]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     from keras import layers
     import pickle as pkl
     from sklearn.preprocessing import LabelEncoder
```

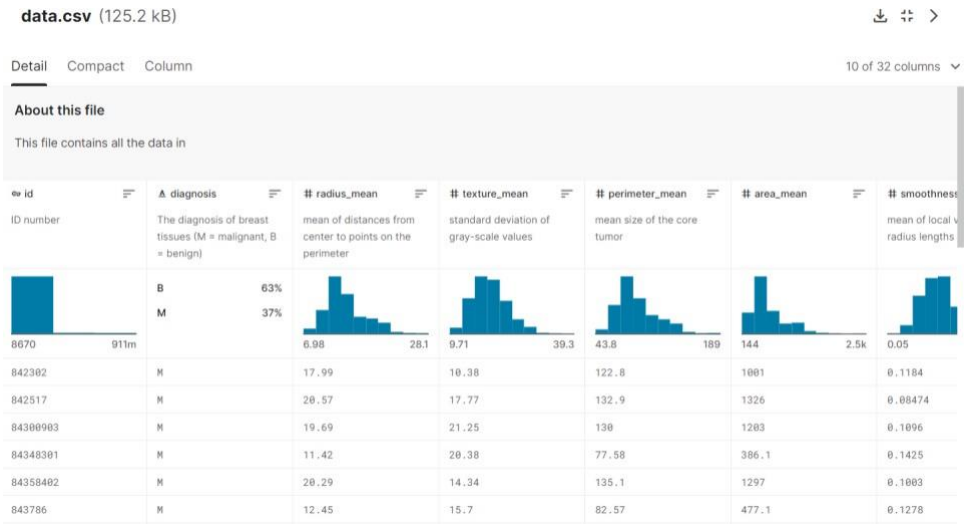
Picture 2.2 - Illustration of imported main modules and libraries

```
]  from keras import layers
   from keras.layers import Dense
   from keras.models import Sequential

   from sklearn.metrics import accuracy_score, confusion_matrix, f1_score
   from sklearn.metrics import classification_report
   from sklearn.model_selection import KFold
   from sklearn.model_selection import cross_validate, cross_val_score
   from sklearn.svm import SVC
   from sklearn import metrics
   from sklearn.linear_model import LogisticRegression
   from sklearn.tree import DecisionTreeClassifier
   from sklearn.ensemble import RandomForestClassifier
   from sklearn.naive_bayes import GaussianNB
   from sklearn.neighbors import KNeighborsClassifier
   from sklearn.model_selection import GridSearchCV
   from sklearn.preprocessing import LabelEncoder
   from sklearn.model_selection import train_test_split
   from sklearn.preprocessing import StandardScaler
```

Picture 2.3 - Illustration of imported other modules and libraries

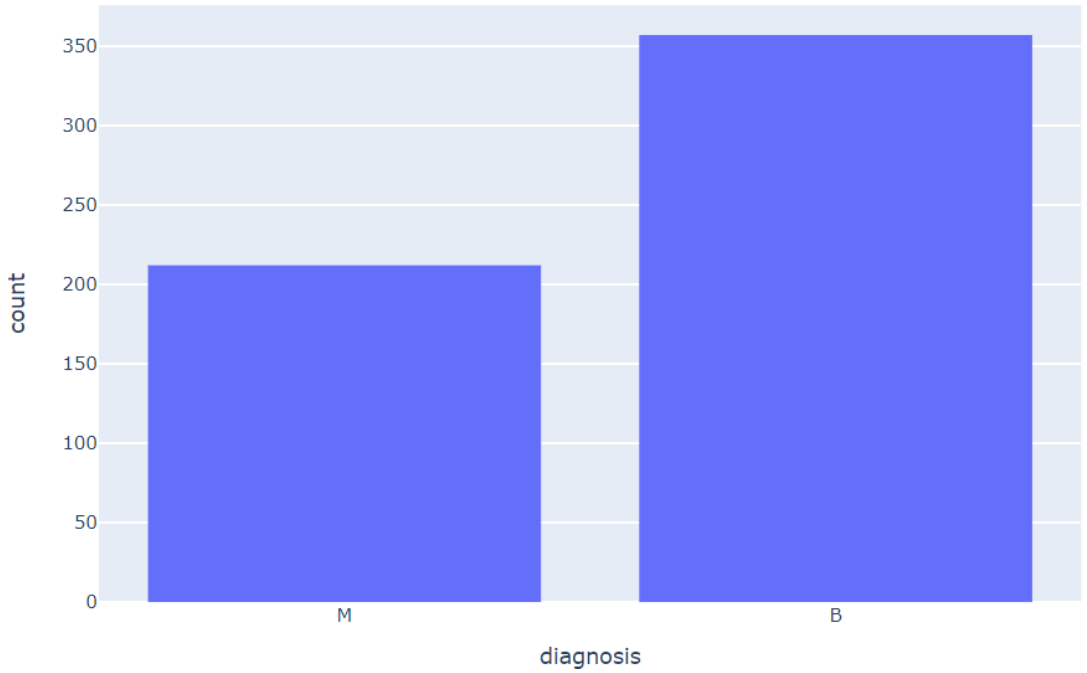
2.2 Dataset features

Open-source dataset from Kaggle has overall 32 features:



Picture 2.4 – Illustration showing dataset

1) Sample ID – it is identification number of patients
2) Classes – M: malignant or B: benign breast mass



Picture 2.5 – The diagram shows us the rates of patients with benign and malignant tumor

Ten features listed below were measured for each cell nucleus:

1) Radius - the average of all distances from the center to the perimeter points
2) Texture – standard deviation of gray-scale values
3) Perimeter
4) Area
5) Smoothness – change in radius length
6) Compactness – formula that sum it

$$C = \frac{P^2}{A - 1}$$

7) Concavity – severity of the contour's concave parts
8) Concave points – the number of concave contour segments
9) Symmetry
10) Fractal dimension – "coastline approximation" – 1

Three measures are provided for each 10 characteristics listed above:

- Mean
- Standard deviation
- Largest – i.e. worst

2.3 Model building

The model building process starts with importing the necessary libraries such as pandas, numpy, scikit-learn, seaborn, keras, and matplotlib. The dataset isthen loaded into a pandas DataFrame, preprocessed, and split into training and testing sets using scikit-learn's train_test_split() function.

```
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, f1_score
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_validate, cross_val_score
from sklearn.svm import SVC
from sklearn import metrics
```

Picture 2.6 – Libraries used in code

The data is then scaled using scikit-learn's StandardScaler() to bring all features to the same scale. After that, the data is encoded using scikit-learn's LabelEncoder() to convert the categorical labels to numerical values so that they may be fitted by machine learning models that only accept numerical input. It is akey stage in the pre-processing of a machine-learning project.

The next step is to build the neural network using Keras. The neural network model consists of an input layer, several hidden layers with ReLU activation functions that adds nonlinearity to a deep learning network and overcomes the vanishing gradients problem, and dropout layers - regularizate approach that prevents overfitting by ensuring that no units are codependent with one another.

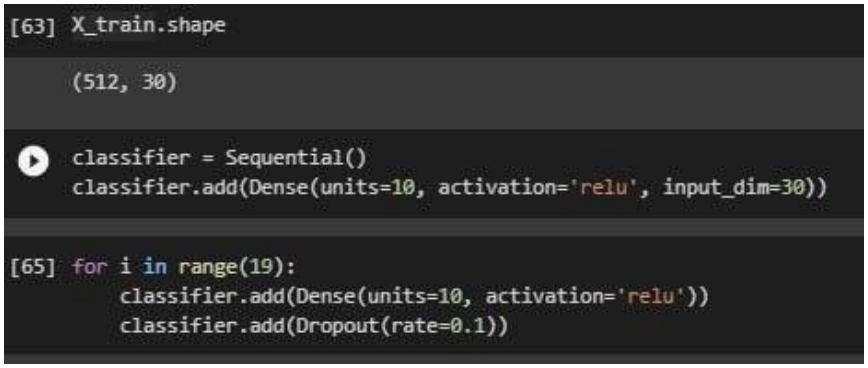
An output layer with a sigmoid activation function that determines which values to pass as output and which to reject. The model is compiled using the ‘adam’ optimizer technique for deep learning model training, the advantages of this optimizer: computational efficiency, low memory requirements, binary cross-entropy loss function that measures the difference between predicted probabilitiesand actual binary labels in classification tasks and accuracy is chosen as the evaluation metric.

To implement machine learning in our project, we used the Sequential model from Keras. The Sequential model is a linear stack of layers.

The first layer is the Dense layer with 10 units and 'relu' activation function. This layer takes an input of 30 features.

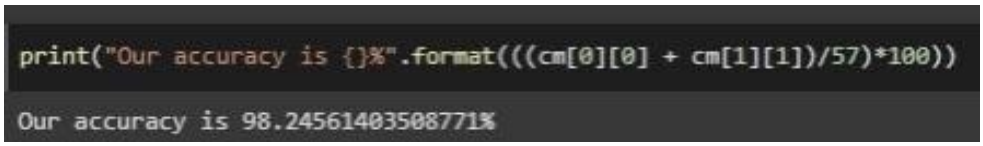
The second layer is a Dropout layer with a dropout rate of 0.1. We added 19 more layers, each with 10 units and 'relu' activation function followed by a Dropout layer with a dropout rate of 0.1. Dropout is used for regularization to prevent overfitting.

In total we used 20 layers with 10 neurons each. (Picture 2.3.2)
The final layer is a Dense layer with 1 unit and 'sigmoid' activation function.
This layer produces a binary output indicating the probability of breast cancer.



Picture 2.7– Number of layers and neurons implemented

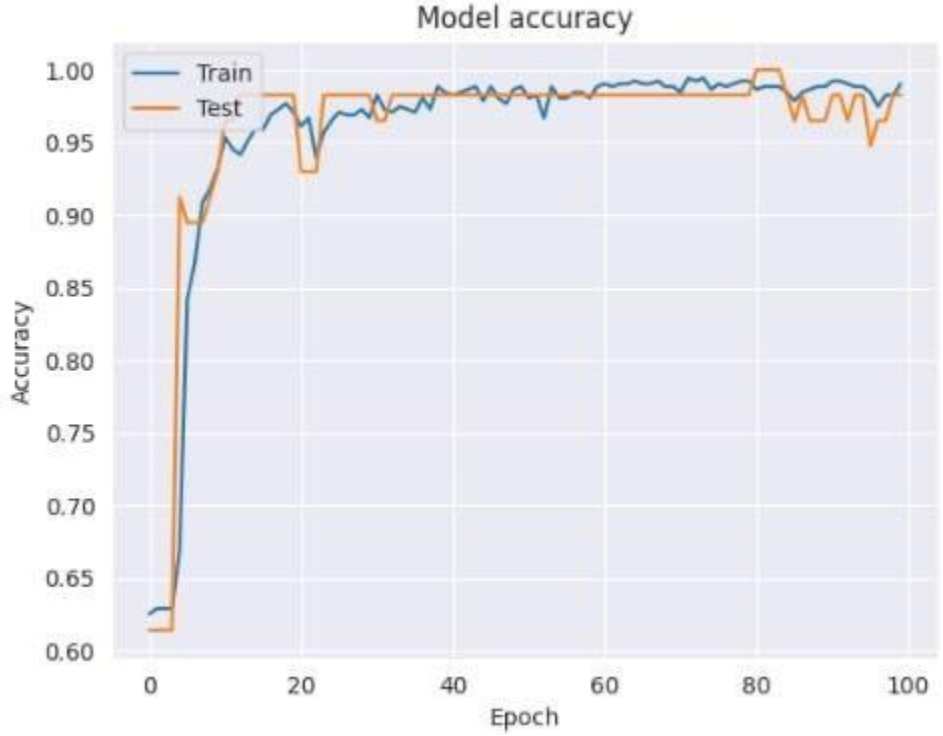
Before training the model, we compiled it using 'adam' optimizer, 'binary_crossentropy' loss function, and 'accuracy' metric.
We trained the model for 100 epochs with a batch size of 16 and validated it using the test dataset. When using epoch = 100 and batch size = 16, we achieved an accuracy of 98.24%. (Picture 2.3.3)



Picture 2.8 – Accuracy got by using batch size = 16 and epoch = 100

The model is trained using the fit() function of Keras with the training data and validated using the validation data. The accuracy and loss of the model are then plotted using matplotlib.

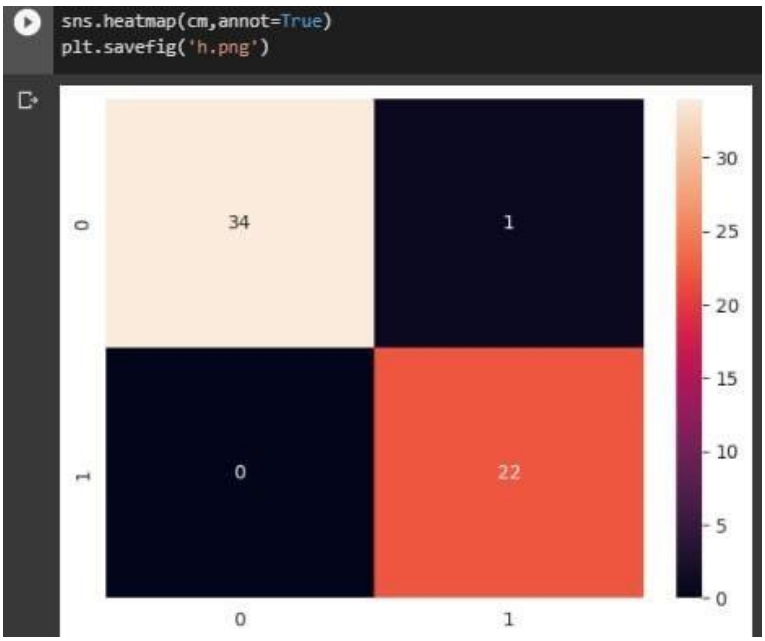
After training, we plotted the model accuracy over the epochs to evaluate the performance of the model. We then used the model to predict the output for the test dataset and calculated the confusion matrix and accuracy score using scikit-learn.



Picture 2.9 – Model accuracy plotted when batch size = 16 and epoch = 100

In the context of breast cancer detection, the process of training a machine learning model involves several steps. Once the dataset is prepared, the model is trained by dividing the data into batches and running them through the model for a given number of epochs. At the end of each epoch, the loss and accuracy values for both the training data and the test data are computed.

Using these values, we plotted graphs to compare the loss and accuracy of the two datasets. With each epoch, the loss and validation loss values gradually decrease and will soon intersect, while the accuracy and validation accuracy values increase and will also soon intersect. When this occurs, it is an indication that the model has been built correctly and that it is performing well.

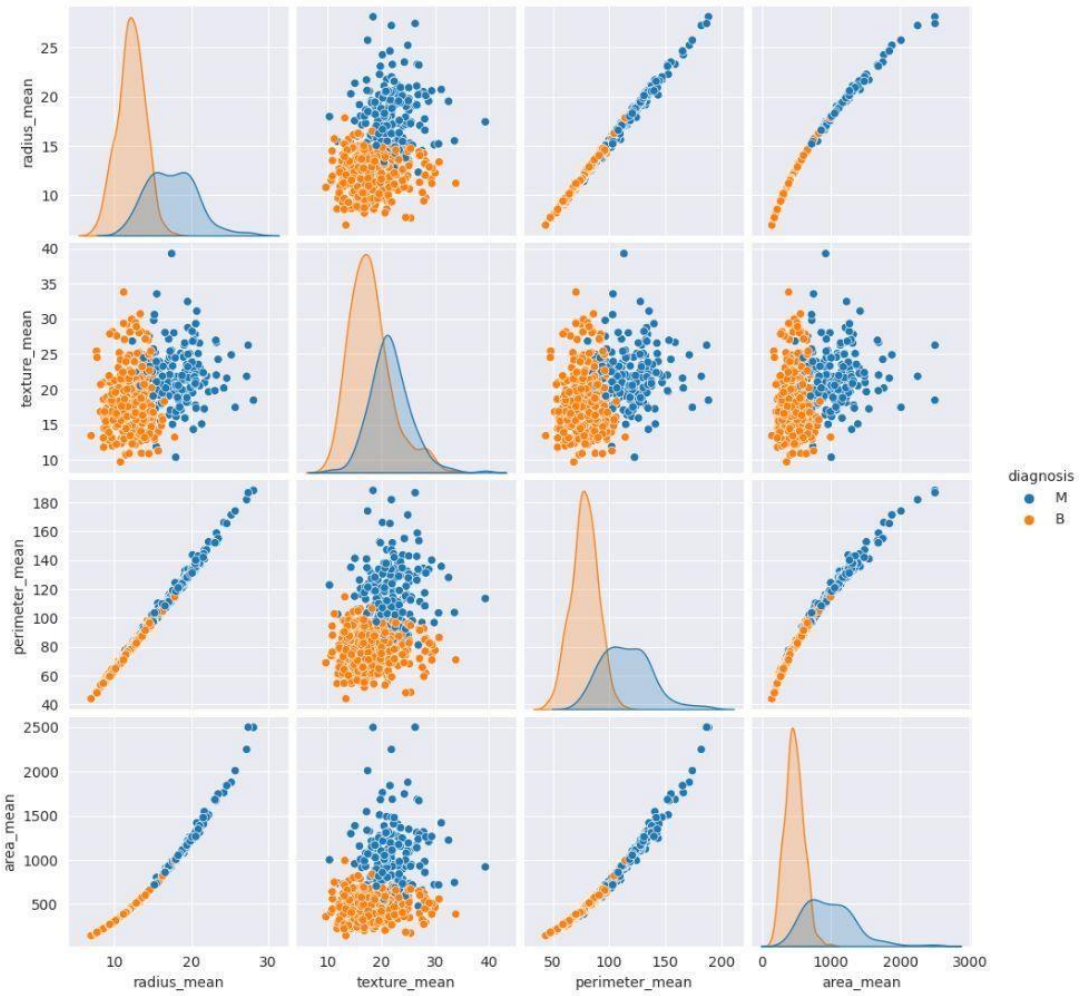


Picture 2.10 – Heatmap of model building

Finally, the model is evaluated using the testing data, and the confusion matrix and accuracy score are calculated using scikit-learn's confusion_matrix() and accuracy_score() functions. The confusion matrix is visualized using seaborn's heatmap() function, and the accuracy score is printed on the console.

3 RESULT OF RESEARCH

In the course of our work, we used a dataset from Kaggle, it contains about 30 columns that describe the disease. For example: the size of the tumor, the type of tumor (malignant or benign), the texture of the tumor, and many other characteristics. According to these characteristics, more than 500 records have been recorded. Next, we deduced the exact number of malignant and benign tumors, which was B(benign)=357, M(malignant)=212. And then plotted this data on a graph.
(Picture 3.1)

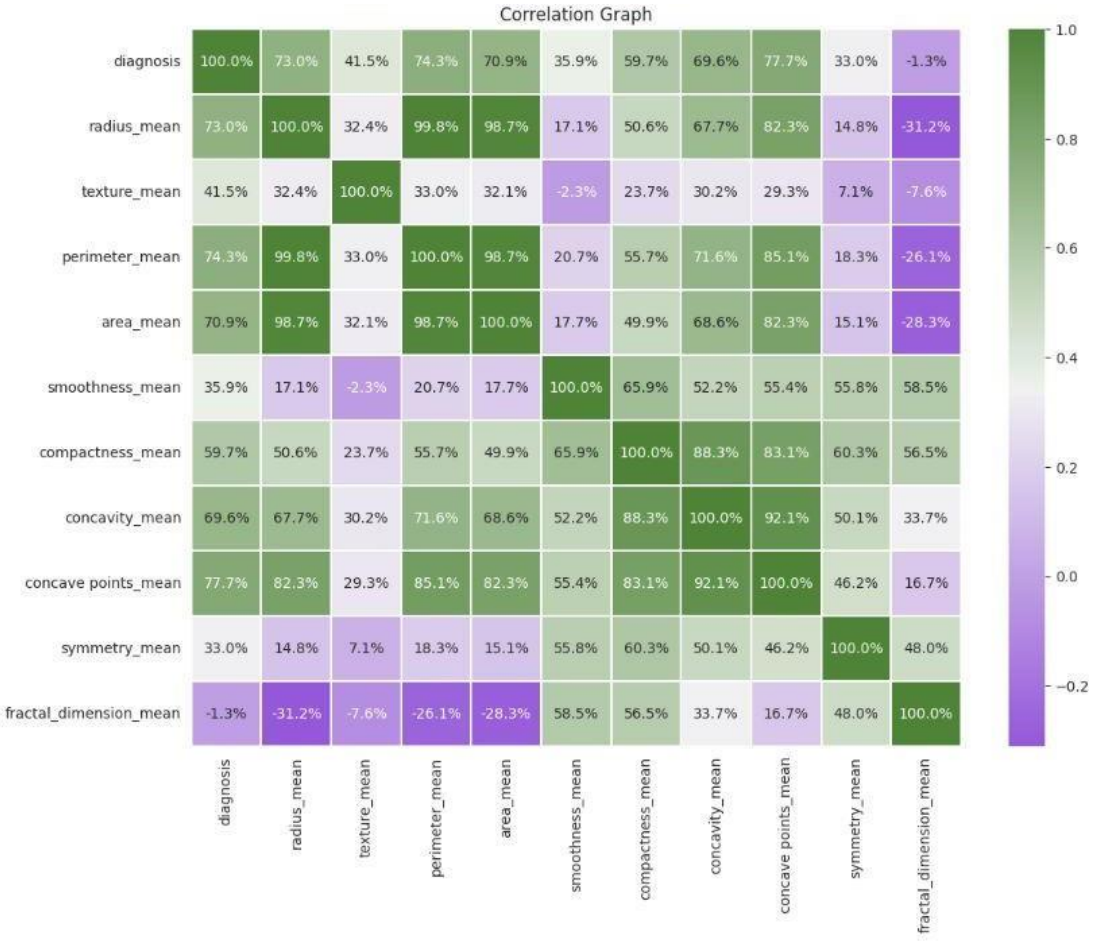


Picture 3.1 - Feature distribution of dataset with samples of (**a**) Radius mean, (**b**) Texture mean, (**c**) Perimeter mean, (**d**) Area mean

Next, we test the correlation between mean features by plotting pairwise relationships between the diagnosis (benign, malignant) and characteristics such as radius, texture, and so on. From the results of these diagrams, we can see that malignant cells tend to have large average values for the characteristics: radius, perimeter area, compactness, concavity and concave points. After plotting the correlation plot, we

confirm that there is a strong correlation between radius, perimeter and area. Compactness, concavity and concavity points also change together. Thanks to the correlation analysis that we conducted, we were able to geta more detailed perception of the dataset, since in the future it will help us to compare already used classifiers. Correlation analysis made us understand the relationship between variables, we identified dependent variables and then removed redundant variables so that there was no distortion of the results.

Therefore, since there is a strong relationship between radius, perimeter and area, as well as between concavity, compactness and concave points, we can remove redundant variables to avoid bias in further predictions. (Picture 3.2)



Picture 3.2 – Correlation Graph determine which features are most correlated with each other with a diagnosis (feature)

After carrying out the correlation analysis, we can begin to separate the data into training (train) and test (test). This is an important step in our future work, since with it we can make predictions on new data, determine the performance of models, and we can also detect model overfitting. Since there are a lot of indicatorsin our dataset that need to be taken into account, we will transform the data so that there is no strong spread between them and that they are in a certain range. We do this with StandardScaler(). (Picture 3.3) That is, we apply it to the training dataset (X_train) to

calculate the scaling parameters (mean and standard deviation) and then apply those same parameters to the test dataset (X_test) to bring it to the samescale. Thus, since all parameters will be reduced to a common level of values, this will improve the performance of machine learning algorithms and help to obtain more accurate results.

```
In [36]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.fit_transform(X_test)
```

Picture 3.3 – Screenshot of code with StandardScaler

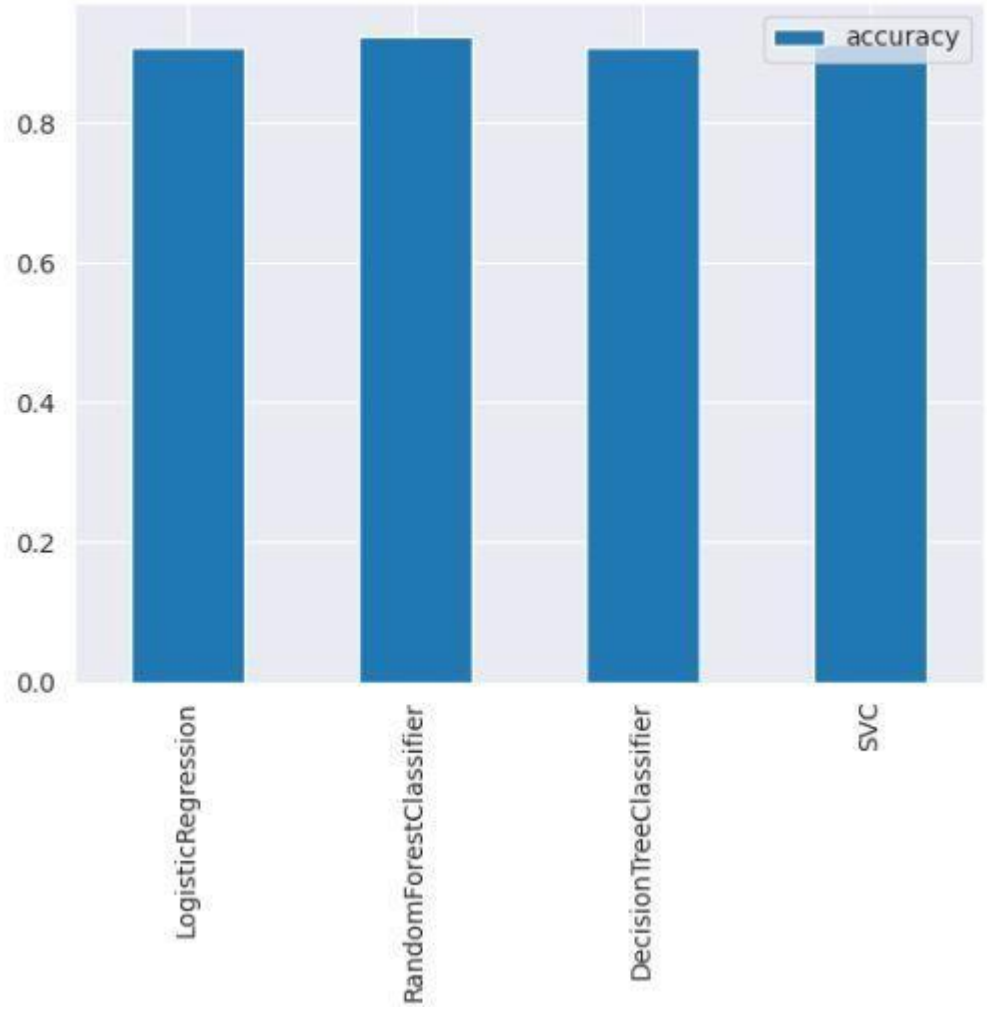
After applying the StandardScaler() to the results and selecting them for the test and test sets, the models are tested and evaluated for their value. To do this, we write the model_building() function, which takes the model, the test and test data, and the class labels of the test set. Inside the function, the model is trained on the training data (model.fit(X_train, y_train)) and its accuracy is evaluated on the training data (score = model.score(X_train, y_train)). The model then makes predictions on tests of the data (predictions = model.predict(X_test)) and calculatesthe accuracy of the predictions using the accuracy metric (precision = accuracy_score(predictions, y_test)). We will now train the models predictably andevaluate the performance among various four machine learning models. They are Logistic Regression, Random Forest Classifier, Decision Tree Classifier and SVC-Support Vector Classifier.

```
In [38]: models_list = {
             "LogisticRegression" :  LogisticRegression(),
             "RandomForestClassifier" :  RandomForestClassifier(n_estimators=10, criterion='entropy', random_state=5),
             "DecisionTreeClassifier" :  DecisionTreeClassifier(criterion='entropy', random_state=0),
             "SVC" :  SVC(),
         }
         print(models_list)
```

Picture 3.4 – Screenshot of code with Classifiers

After comparing the models, we proceed to evaluate the performance of machine learning classification models. To do this, we use a matrix of errors (confusion matrix) and a heat map (heatmap), which, based on the matrix of errors,visualizes the values for us. It represents each value in the error matrix with a color, with higher values in bright color and lower values in dim color. Such visualization helps to quickly assess which classes the model misclassifies and how often this happens. (Picture 3.5)

```
In [41]: model_compare = pd.DataFrame(scores, index=["accuracy"])
         model_compare.T.plot.bar();
```

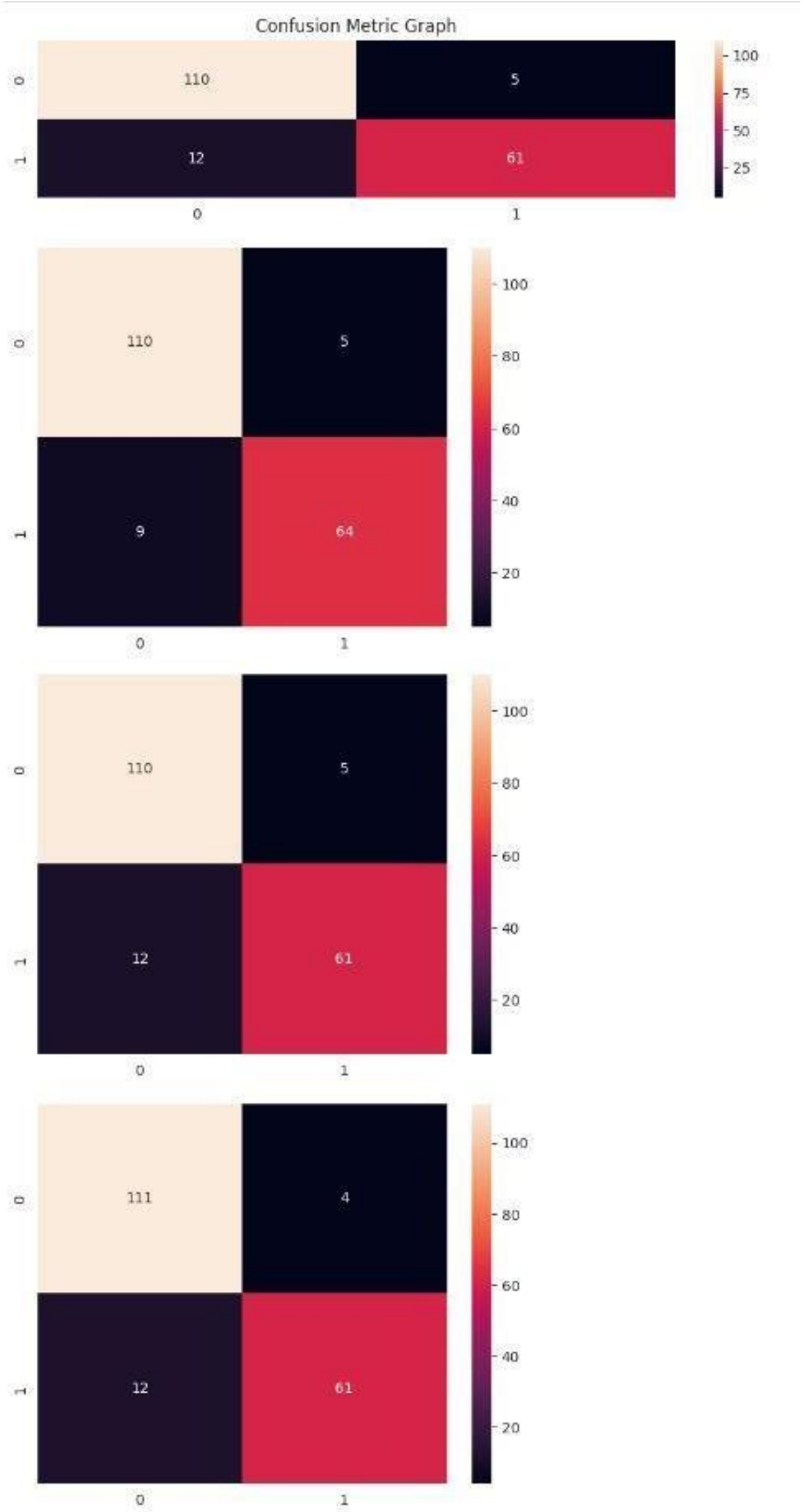


Picture 3.5 - Comparative Graph of Classifier Accuracy

By building our classification model, we can see that the randomized forest classification algorithm produces the best results for our dataset. However, when using deep learning with neural networks – ANN, we can get even better model performance and prediction accuracy. With deep learning, the model achieved a prediction accuracy of 98.24%.

We used 20 layers of 10 neurons each, increased the number of epochs and as a result we have excellent accuracy, which did not fall into overfitting.

This indicates the high performance and reliability of our classification model, making it a suitable choice for solving the classification problem on this dataset.



Picture 3.6 – Confusion Metric Graph

CONCLUSION

Accurate and timely detection of various diseases, such as breast cancer, is still an important issue in healthcare. In this study, five-level master data exploration approaches using five different machine learning prediction models, including SVM, logistic regression, KNN, random forest, and decision tree classifier, were proposed to identify and classify breast cancer tumors.

Meanwhile, the random forest achieved an accuracy of 92.55%, demonstrating that machine learning algorithms may identify higher precision. Our findings demonstrate the accuracy of our prediction models for detecting breast cancer while requiring only a short model training period of time. These complex models, methodologies, and findings will assist physicians and data analysts in using a more intelligent classifier to identify breast cancer symptoms.

Since breast cancer-related picture data is accessible, we applied deep learning models for breast cancer diagnosis - ANN, which resulted in an excellent outcome of 100 epochs with 98% accuracy.

During the analysis, we focused on how the model was trained and how the data sets matched the final outcome. We tried several options and came to the conclusion that this model can predict the type of breast cancer tumor from the data with 98% accuracy, which is an excellent result. This model can serve as an assistant for doctors, analysts, it can also be modified for datasets related to medicine.

REFERENCES

[1] Prasad, Y.; Biswas, K.K.; Jain, C.K., SVM classifier-based feature selection using GA, ACO and PSO for RNA design.
Available: https://link.springer.com/chapter/10.1007/978-3-642-13498-2_40

[2] De Moulin D., "A short history of breast cancer", Boston: Martinus Nijhoff; 1989, p. 1-133.

[3] Garrison FH, "An introduction to the history of medicine. 4th ed", Philadelphia: WB Saunders, Co., 1929. p. 588–9.

[4] Encyclopedia Britannica, 15th ed. Chicago: Encyclopedia Britannica Inc., 1978. Macropedia Vol 11 p. 837.

[5] Müller J., "Uber den feinen Bau und der Formen dei Krankhafte Geschwülste", Berlin: G Reimer, 1838.

[6] Bloom, HJG, Richardson WW, Harries EJ., "Natural history of untreated breast cancer (1805-1933)", Comparison of untreated and treated cases according to histological grade of malignancy, Brit Med J 1962, I:213–21.

[7] Donegan WL.," Staging and prognosis", In: Donegan WL, Spratt JS, editors, Cancer of the breast, 5th ed. Philadelphia: W.B. Saunders Co, 2002. p. 478.

[8] Wolfe JN., "Xeroradiography of the breast", Springfield: Charles C Thomas, 1972. p. 3–5.

[9] DeVita VT., "Principles of chemotherapy", In: DeVita VT Jr, Hellman S, Rosenberg SA, editors, Cancer—principles and practice of oncology, Philadelphia: J. B. Lippincott Co, 1982. p. 132–3.

[10] В Казахстане женщины болеют раком почти в 1,5 раза чаще мужчин
Available: https://pkzsk.info/v-kazakhstane-zhenshhiny-boleyut-rakom

[11] Как машинное обучение помогает в борьбе с онкологией
Available: https://trends.rbc.ru/trends/industry

[12] AI Outperforms Radiologists for Detecting Breast Cancer on Mammography
Available: https://web.archive.org/web/20200330013351

[13] Houssami N., Kirkpatrick-Jones G., Noguchi N., "Artificial Intelligence (AI) for the early detection of breast cancer: a scoping review to assess AI's potential in breast screening practice", Expert Rev Med Devices, 2019., Vol. 16.

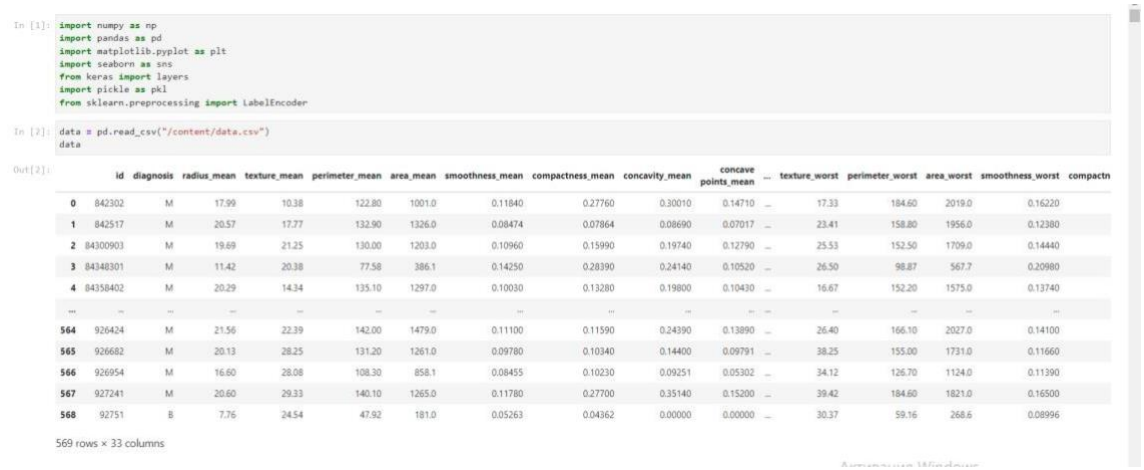
[14] Deep learning-based cardiovascular image diagnosis: A promising challenge
Available: https://www.sciencedirect.com/science/article/abs

[15] Limitations of Mammograms
Available: https://www.cancer.org/cancer

[16] SUPERVISED AND UNSUPERVISED LEARNING
Available: https://www.ibm.com/cloud/blog/supervised

[17] LOGISTIC REGRESSION
Available: https://www.ibm.com/topics/logistic-regression
[18] DECISION TREE
Available: https://www.ibm.com/topics/decision-trees
[19] RANDOM FOREST
Available: https://towardsdatascience.com/understanding-random-forest
[20] SUPPORT VECTOR MACHINE
Available: https://www.ibm.com/topics/decision-trees
[21] GAUSSIAN NAÏVE BAYES
Available: https://pub.towardsai.net/gaussian-naive-bayes
[22] K-NEAREST NEIGHBORS
Available: https://proglib.io/p/metod-k
[23] ARTIFICIAL NEURAL NETWORKS
Available: https://www.ibm.com/topics/neural-networks
[24] ABOUT GOOGLE COLAB
Available: https://colab.research.google.com/
[25] SCIKIT-LEARN
Available: https://www.codecademy.com/article/scikit-learn
[26] NUMPY
Available: https://numpy.org/
[27] PANDAS
Available: https://khashtamov.com/ru/pandas-introduction/
[28] MATPLOTLIB
Available: https://matplotlib.org/
[29] PICKLE
Available: https://www.datacamp.com/tutorial/pickle-python-tutorial#what-
[30] SEABORN
Available: https://pythonru.com/biblioteki/seaborn-plot
[31] KERAS
Available: https://keras.io/

APPENDIX



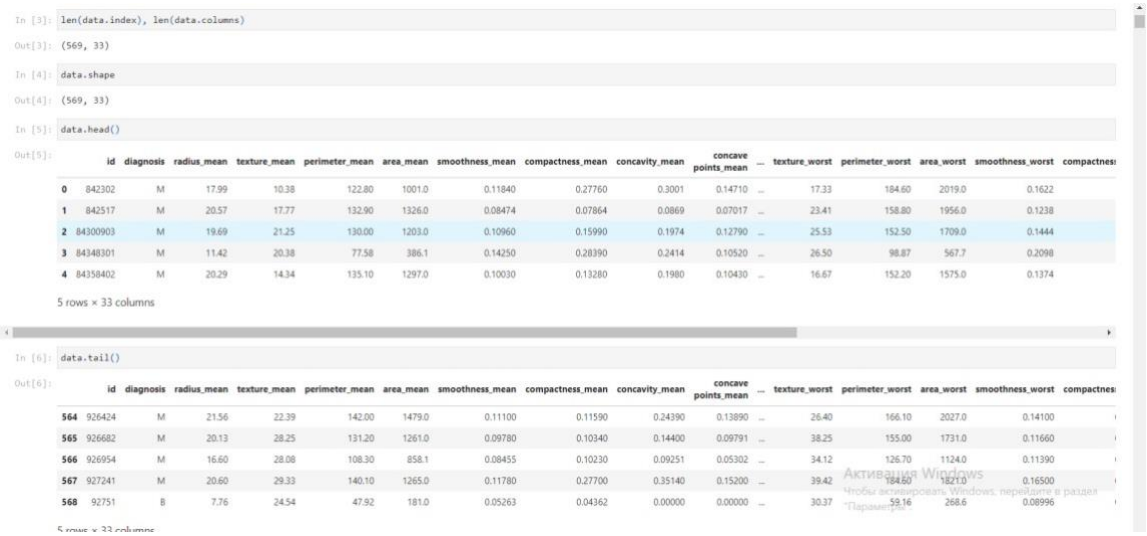
```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from keras import layers
        import pickle as pkl
        from sklearn.preprocessing import LabelEncoder

In [2]: data = pd.read_csv("/content/data.csv")
        data
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactn
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	17.33	184.60	2019.0	0.16220	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	23.41	158.80	1956.0	0.12380	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	25.53	152.50	1709.0	0.14440	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	26.50	98.87	567.7	0.20980	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	16.67	152.20	1575.0	0.13740	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	26.40	166.10	2027.0	0.14100	
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	38.25	155.00	1731.0	0.11660	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	34.12	126.70	1124.0	0.11390	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	39.42	184.60	1821.0	0.16500	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	30.37	59.16	268.6	0.08996	

569 rows × 33 columns

Picture 3.7 – Importing libraries, dataset



```
In [3]: len(data.index), len(data.columns)
Out[3]: (569, 33)
In [4]: data.shape
Out[4]: (569, 33)
In [5]: data.head()
```

Out[5]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactnes
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	17.33	184.60	2019.0	0.1622	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	23.41	158.80	1956.0	0.1238	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	25.53	152.50	1709.0	0.1444	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	26.50	98.87	567.7	0.2098	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	16.67	152.20	1575.0	0.1374	

5 rows × 33 columns

```
In [6]: data.tail()
```

Out[6]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactnes
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	26.40	166.10	2027.0	0.14100	
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	38.25	155.00	1731.0	0.11660	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	34.12	126.70	1124.0	0.11390	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	39.42	184.60	1821.0	0.16500	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	30.37	59.16	268.6	0.08996	

5 rows × 33 columns

Picture 3.8 – Researching dataset

```
In [11]: data = data.dropna(axis='columns')

In [12]: data.describe(include="O")
```

Out[12]:

	diagnosis
count	569
unique	2
top	B
freq	357

```
In [13]: data.diagnosis.value_counts()

Out[13]: B    357
         M    212
         Name: diagnosis, dtype: int64

In [14]: data.head(2)
```

Out[14]:

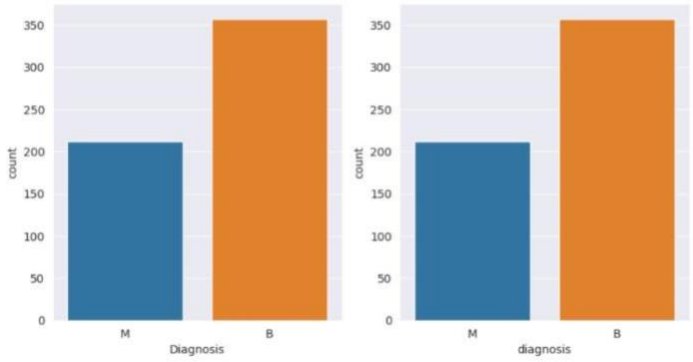
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
0	842302	M	17.99	10.38	122.8	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	184.6	2019.0	0.1622
1	842517	M	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	158.8	1956.0	0.1238

2 rows × 32 columns

```
In [15]: diagnosis_unique = data.diagnosis.unique()

In [16]: diagnosis_unique

Out[16]: array(['M', 'B'], dtype=object)
```


Picture 3.9 – Unique values

```
In [17]: import plotly.express as px
         import plotly.graph_objects as go

         %matplotlib inline
         sns.set_style('darkgrid')

In [18]: plt.figure(figsize=(10,5))
         plt.subplot(1, 2, 1)
         sns.countplot(x='diagnosis', data=data)
         plt.xlabel("Diagnosis")
         plt.subplot(1, 2, 2)
         sns.countplot(x='diagnosis', data=data)
         plt.show()
```



Picture 3.10 – Graph showing values of malignant and benign

```
[ ] cols = ['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
            'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
            'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean']
    print(len(cols))
    data[cols].corr()
```

11

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_
diagnosis	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360	0.776614	0.330499	-0.0
radius_mean	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529	0.147741	-0.3
texture_mean	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464	0.071401	-0.0
perimeter_mean	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977	0.183027	-0.2
area_mean	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269	0.151293	-0.2
smoothness_mean	0.358560	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553695	0.557775	0.5
compactness_mean	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135	0.602641	0.5
concavity_mean	0.696360	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391	0.500667	0.3
concave points_mean	0.776614	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000000	0.462497	0.1
symmetry_mean	0.330499	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497	1.000000	0.4
fractal_dimension_mean	-0.012838	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166917	0.479921	1.0

Picture 3.11 – Correlation graph

```
def model_building(model, X_train, X_test, y_train, y_test):

    model.fit(X_train, y_train)
    score = model.score(X_train, y_train)
    predictions = model.predict(X_test)
    accuracy = accuracy_score(predictions, y_test)

    return (score, accuracy, predictions)
```

```
models_list = {
    "LogisticRegression" :  LogisticRegression(),
    "RandomForestClassifier" :  RandomForestClassifier(n_estimators=10, criterion='entropy', random_state=5),
    "DecisionTreeClassifier" :  DecisionTreeClassifier(criterion='entropy', random_state=0),
    "SVC" :  SVC(),
}
print(models_list)
```

```
{'LogisticRegression': LogisticRegression(), 'RandomForestClassifier': RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=5), 'DecisionTreeClassifier': Decisior
```

```
[ ] def train_score(models, X_train, X_test, y_train, y_test):
      np.random.seed(0)
      scores = {}
      for name, model in models.items():
          model.fit(X_train, y_train)
          scores[name] = model.score(X_test, y_test)
      return scores
```


Picture 3.12 – Model building

```
df_prediction = []
confusion_matrixs = []
df_prediction_cols = [ 'model_name', 'score', 'accuracy_score' , "accuracy_percentage"]
for name, model in zip(list(models_list.keys()), list(models_list.values())):
    (score, accuracy, predictions) = model_building(model, X_train, X_test, y_train, y_test )
    print("\n\nClassification Report of '"+ str(name), "'\n")
    print(classification_report(y_test, predictions))
    df_prediction.append([name, score, accuracy, "{0:.2%}".format(accuracy)])
    confusion_matrixs.append(confusion_matrix(y_test, predictions))

df_pred = pd.DataFrame(df_prediction, columns=df_prediction_cols)
```

```
Classification Report of 'LogisticRegression '

              precision    recall  f1-score   support

           0       0.90      0.96      0.93       115
           1       0.92      0.84      0.88        73

    accuracy                           0.91       188
   macro avg       0.91      0.90      0.90       188
weighted avg       0.91      0.91      0.91       188
```

Picture 3.13 – Classification Report (Part 1)

```
Classification Report of 'RandomForestClassifier '

              precision    recall  f1-score   support

           0       0.92      0.96      0.94       115
           1       0.93      0.88      0.90        73

    accuracy                           0.93       188
   macro avg       0.93      0.92      0.92       188
weighted avg       0.93      0.93      0.93       188
```

Picture 3.13 – Classification Report (Part 2)

```
Classification Report of 'DecisionTreeClassifier '

              precision    recall  f1-score   support

           0       0.90      0.96      0.93       115
           1       0.92      0.84      0.88        73

    accuracy                           0.91       188
   macro avg       0.91      0.90      0.90       188
weighted avg       0.91      0.91      0.91       188
```

Picture 3.13 – Classification Report (Part 3)

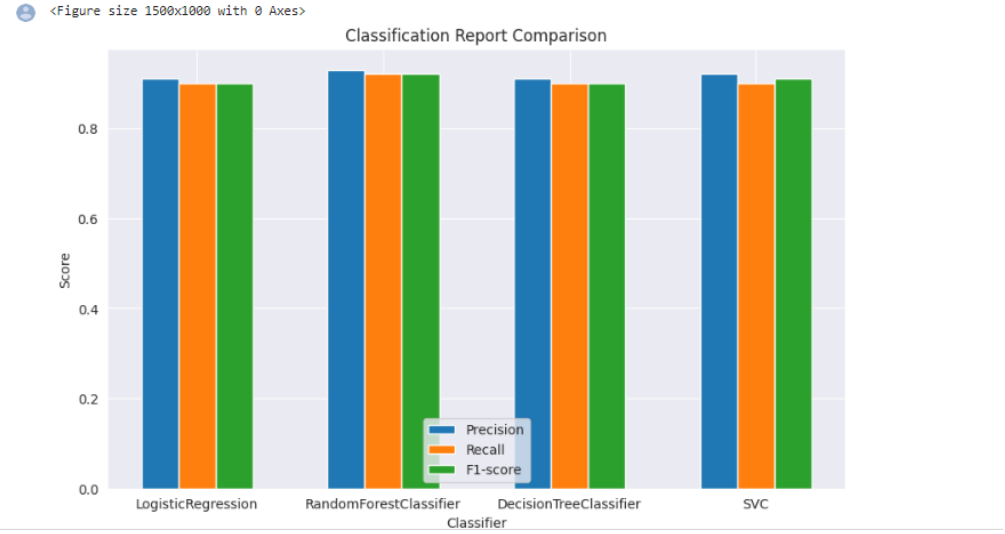
Classification Report of 'SVC '

	precision	recall	f1-score	support
0	0.90	0.97	0.93	115
1	0.94	0.84	0.88	73
accuracy			0.91	188
macro avg	0.92	0.90	0.91	188
weighted avg	0.92	0.91	0.91	188

Picture 3.13 – Classification Report (Part 4)

```
plt.figure(figsize=(15, 10))
fig = px.imshow(data[cols].corr());
classifiers = ['LogisticRegression', 'RandomForestClassifier', 'DecisionTreeClassifier', 'SVC']
precision = [0.91, 0.93, 0.91, 0.92]
recall = [0.90, 0.92, 0.90, 0.90]
f1_score = [0.90, 0.92, 0.90, 0.91]
x_ticks = np.arange(len(classifiers))
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x_ticks - 0.2, precision, 0.2, label='Precision')
ax.bar(x_ticks, recall, 0.2, label='Recall')
ax.bar(x_ticks + 0.2, f1_score, 0.2, label='F1-score')
ax.set_xlabel('Classifier')
ax.set_ylabel('Score')
ax.set_title('Classification Report Comparison')
ax.set_xticks(x_ticks)
ax.set_xticklabels(classifiers)
ax.legend()
plt.show()
fig.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```



Picture 3.14 – Classification Report in Graph

```
df_pred.sort_values('score', ascending=False)
df_pred.sort_values('accuracy_score', ascending=False)
```

	model_name	score	accuracy_score	accuracy_percentage
1	RandomForestClassifier	0.992126	0.925532	92.55%
3	SVC	0.923885	0.914894	91.49%
0	LogisticRegression	0.916010	0.909574	90.96%
2	DecisionTreeClassifier	1.000000	0.909574	90.96%

Picture 3.15 – Accuracy scores

```
for name, model in zip(list(models_list.keys()), list(models_list.values())):
    cross_val_scorring(model)
```

```
Full-Data Accuracy: 0.9
Cross Validation Score of'LogisticRegression '

Score: 0.91
Score: 0.91
Score: 0.9
Score: 0.9
Score: 0.9

Full-Data Accuracy: 1.0
Cross Validation Score of'RandomForestClassifier '

Score: 0.99
Score: 0.99
Score: 0.99
Score: 1.0
Score: 1.0

Full-Data Accuracy: 1.0
Cross Validation Score of'DecisionTreeClassifier '

Score: 1.0
Score: 1.0
Score: 1.0
Score: 1.0
Score: 1.0

Full-Data Accuracy: 0.89
Cross Validation Score of'SVC '

Score: 0.9
Score: 0.89
Score: 0.88
Score: 0.88
Score: 0.88
```

Picture 3.16 – Cross Validation for score

```
[ ] model = KNeighborsClassifier()

    param_grid = {
        'n_neighbors': list(range(1, 30)),
        'leaf_size': list(range(1,30)),
        'weights': [ 'distance', 'uniform' ]
    }

    gsc = GridSearchCV(model, param_grid, cv=10)
    gsc.fit(X_train, y_train)

    print("\n Best Score is ")
    print(gsc.best_score_)

    print("\n Best Estinator is ")
    print(gsc.best_estimator_)

    print("\n Best Parametes are")
    print(gsc.best_params_)
```

```
 Best Score is
0.9159244264507423

 Best Estinator is
KNeighborsClassifier(leaf_size=1, n_neighbors=10)

 Best Parametes are
{'leaf_size': 1, 'n_neighbors': 10, 'weights': 'uniform'}
```

Picture 3.17 – KNN model

```
model = SVC()
param_grid = [
              {'C': [1, 10, 100, 1000],
               'kernel': ['linear']
              },
              {'C': [1, 10, 100, 1000],
               'gamma': [0.001, 0.0001],
               'kernel': ['rbf']
              }
]

gsc = GridSearchCV(model, param_grid, cv=10) # 10 Cross Validation
gsc.fit(X_train, y_train)

print("\n Best Score is ")
print(gsc.best_score_)

print("\n Best Estinator is ")
print(gsc.best_estimator_)

print("\n Best Parametes are")
print(gsc.best_params_)
```

```
 Best Score is
0.9184885290148447

 Best Estinator is
SVC(C=10, gamma=0.001)

 Best Parametes are
{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
```

Picture 3.18 – SVM Model

```
 Best Score is
0.9132253711201079

 Best Estinator is
RandomForestClassifier(max_depth=40, max_features='auto', min_samples_leaf=2,
                       n_estimators=200)

 Best Parametes are
{'bootstrap': True, 'max_depth': 40, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
```

Picture 3.19 – Random Forest Model

```
 Best Score is
0.9265856950067477

 Best Estinator is
DecisionTreeClassifier(max_features='log2', min_samples_leaf=6,
                       min_samples_split=9)

 Best Parametes are
{'max_features': 'log2', 'min_samples_leaf': 6, 'min_samples_split': 9}
```

Picture 3.20 – Decision Tree Model

```
[ ]  logistic_model = LogisticRegression()
     logistic_model.fit(X_train, y_train)

     filename = 'logistic_model.pkl'
     pkl.dump(logistic_model, open(filename, 'wb'))
```

```
[ ]  loaded_model = pkl.load(open(filename, 'rb'))
     result = loaded_model.score(X_test, y_test)
```

```
[ ]  result

     0.9095744680851063
```

Picture 3.21 – Logistic Regression Model

```
from keras.layers import Dropout
from keras.layers import Dense
from keras.models import Sequential

X = data.iloc[:, 2:].values
y = data.iloc[:, 1].values

labelencoder_X_1 = LabelEncoder()
y = labelencoder_X_1.fit_transform(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
X_train.shape
```

```
(512, 30)
```

```
classifier = Sequential()
classifier.add(Dense(units=10, activation='relu', input_dim=30))
```

```
for i in range(19):
    classifier.add(Dense(units=10, activation='relu'))
    classifier.add(Dropout(rate=0.1))
```

```
classifier.add(Dense(units=1, activation='sigmoid'))
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

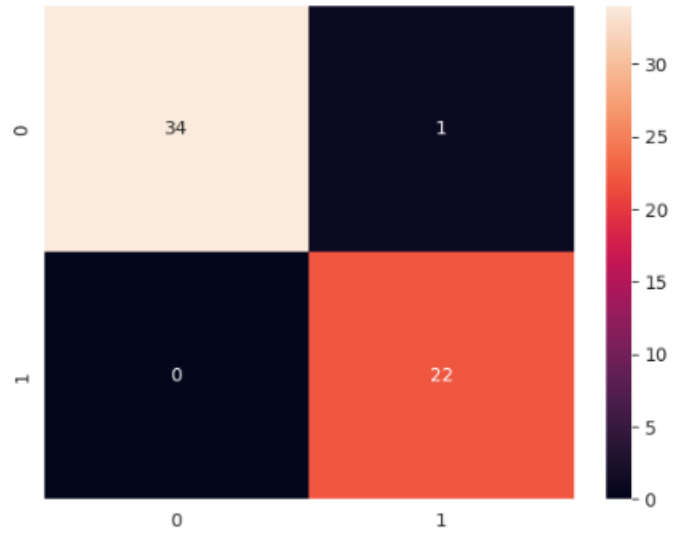
```
history = classifier.fit(X_train, y_train, batch_size=16, epochs=100, validation_data=(X_test, y_test))
```

Picture 3.22 – Keras functions to implement ANN

```
Epoch 96/100
32/32 [==============================] - 0s 8ms/step - loss: 0.0675 - accuracy: 0.9844 - val_loss: 0.2304 - val_accuracy: 0.9474
Epoch 97/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1197 - accuracy: 0.9746 - val_loss: 0.0337 - val_accuracy: 0.9649
Epoch 98/100
32/32 [==============================] - 0s 12ms/step - loss: 0.0722 - accuracy: 0.9824 - val_loss: 0.0549 - val_accuracy: 0.9649
Epoch 99/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1067 - accuracy: 0.9824 - val_loss: 0.0465 - val_accuracy: 0.9825
Epoch 100/100
32/32 [==============================] - 0s 12ms/step - loss: 0.0542 - accuracy: 0.9902 - val_loss: 0.0301 - val_accuracy: 0.9825
```

Picture 3.23 – Results after training 100 epoch, accuracy – 98%

```
[ ] cm = confusion_matrix(y_test, y_pred)
    cm

    array([[34,  1],
           [ 0, 22]])

    print("Our accuracy is {}%".format(((cm[0][0] + cm[1][1])/57)*100))

    Our accuracy is 98.24561403508771%

[ ] sns.heatmap(cm,annot=True)
    plt.savefig('h.png')
```



Picture 3.24 – Heatmap matrix of model building