A Project Report

On

Cancer Prediction using Machine Learning

Submitted in partial fulfillment of the

requirement for the award of the degree of

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

MASTER OF COMPUTER APPLICATION

Session 2023-24

In

Computer Science and Engineering

By

Saksham Gupta (23SCSE2030146)

Rishabh Kumar (23SCSE2030387)

Md. Shabaz Hassan (23SCSE2030358)

Under the guidance of

Dr. Lalit Kumar (Associate Professor)

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

UTTAR PRADESH, INDIA

Jan, 2024

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY
GALGOTIAS UNIVERSITY, GREATER NOIDA,
UTTAR PRADESH, 203201

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"Cancer Prediction Using Machine Learning"** in partial fulfillment of the requirements for the award of the MCA (Master of Computer Application) submitted in the School of Computer Application and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2023 to Jan and 2024, under the supervision of **Dr. Lalit Kumar**, Department of Computer Science and Engineering/School of Computer Application and Technology , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Saksham Gupta (23SCSE2030146)
Rishabh Kumar (23SCSE2030448)
Md.Shahbaz Hassan (23SCSE2030310)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Lalit Kumar
(Associate Professor)

GALGOTIAS UNIVERSITY

Department of School of Computing Science & Engineering

CERTIFICATE

This is to certify that the mini project report entitled "**Cancer Prediction Using Machine Learning**" submitted by Saksham Gupta, Rishabh Kumar, Md. Shahbaz Hassan have been carried out under the guidance of Dr. Lalit Kumar(Associate Professor), Computer Science & Engineering, Galgotias University. The project report is approved for submission requirement for practical examination in 1st semester(MCA) in Computer Science & Engineering from Galgotias University.

SIGNATURE:

Date:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible and their constant encouragement, guidance has been a source of inspiration throughout the course of the project.

We express our sincere indebtedness towards our Prof. Lalit Kumar, Computer Science & Engineering, Galgotias University, for his invaluable guidance, suggestions and supervision throughout the work. Without his kind patronage and guidance, the project would not have taken shape. We would also like to express our gratitude and sincere regards for his kind approval of the project, time to time counselling and advices.

DATE:

Saksham Gupta (23SCSE2030146)

Rishabh Kumar (23SCSE2030448)

Md.Shahbaz Hassan (23SCSE2030310)

ABSTRACT

Area/Domain of Project: Machine Learning

Machine learning is increasingly being employed in cancer detection and diagnosis. Cancer prediction will become quite easy in the future and we can predict it without the need of going to the hospitals. As we can see many technologies are being used and tested in the medical field. So, by this we can say that this will make us easier in the future to detect cancer. We are testing which algorithm will give us good result among CART, SVM AND KNN. We are making a cancer prediction using machine learning, in which we are including three types of cancer they are breast cancer, lungs cancer and prostate cancer. In breast cancer, we are using SVM algorithm and for lung and prostate we are using Random forest algorithm. We are going to give different attributes for three cancer system where the user has to enter data to get result. For breast cancer we are considering attributes like clump thickness, uniform cell size, uniform cell shape etc. and the prediction result will be whether the cancer is malignant or benign. For lung cancer, we are considering smoking, yellow fingers, anxiety, peer pressure etc. In prostate cancer, we are considering are radius, texture, perimeter, area etc. and the result for both cancer is likelihood of being affected by the cancer.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Breast Cancer Prediction Using Machine Learning source code aims to employ a Logistic Regression model to predict whether a breast tumor is malignant or benign based on various features. Leveraging the Breast Cancer dataset provided by the sklearn library, this project is designed to contribute to the early detection and diagnosis of breast cancer, a critical step in improving patient outcomes.

The Logistic Regression algorithm, chosen for its simplicity and effectiveness in binary classification tasks, is employed to analyze patterns within the dataset. The primary objective is to train the model on a subset of the data and subsequently evaluate its performance on both training and testing datasets. The accuracy of the model serves as a crucial metric, reflecting its ability to generalize and make accurate predictions on unseen data.

This project not only focuses on model training and evaluation but also demonstrates the practical application of the trained model through a predictive system. An example input is provided, showcasing how the model can predict whether a given set of tumor characteristics corresponds to a malignant or benign condition. The results obtained from this predictive system contribute to a deeper understanding of the model's real-world utility in assisting medical professionals in diagnosing breast cancer.

By the end of this project, the goal is to have a well-trained Logistic Regression model that can make accurate predictions regarding the nature of breast tumors, thereby aiding in the early identification of potential malignancies and improving the overall effectiveness of breast cancer diagnosis.

CHAPTER 2

LITERATURE SURVEY

1. Machine Learning in Breast Cancer Prediction:

Numerous studies have explored the application of machine learning algorithms for breast cancer prediction. Logistic Regression, Support Vector Machines, Decision Trees, and Neural Networks are commonly employed for their effectiveness in binary classification tasks.

2. Feature Importance and Selection:

Literature emphasizes the importance of selecting relevant features for accurate breast cancer prediction. Researchers often employ feature selection techniques to identify the most informative attributes from the dataset.

3. Integration of Clinical Data:

Some studies integrate clinical data, such as patient history and genetic information, with machine learning models to enhance predictive accuracy. This integration aids in creating more comprehensive models for breast cancer diagnosis.

4. Ensemble Learning Approaches:

Ensemble learning methods, where multiple models are combined to improve overall performance, have gained attention. Random Forests, Gradient Boosting, and Bagging techniques are explored for their potential in breast cancer prediction.

5. Challenges and Future Directions:

Literature discusses challenges, such as dataset imbalance, interpretability of models, and the need for large and diverse datasets. Future research directions focus on addressing these challenges and improving the overall reliability of breast cancer prediction models.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

1. Programming Language:

The code is written in Python, making Python the primary programming language for executing the script.

2. Libraries and Frameworks:

The code relies on various Python libraries and frameworks, including:

NumPy: For numerical operations and array manipulation.

Matplotlib: For data visualization, especially plotting graphs.

Pandas: For data manipulation and analysis.

Seaborn: Enhances the aesthetics of Matplotlib plots.

Scikit-learn: Provides tools for machine learning, including datasets, model training, and evaluation.

3. Development Environment:

Any standard Python development environment, such as Jupyter Notebooks, Spyder, or VSCode, can be used to execute and modify the code.

4. Data Source:

The code utilizes the Breast Cancer dataset from the scikit-learn library. Ensure that scikit-learn is installed, or consider using other data sources if needed.

5. Version Control:

Git: It's advisable to use version control, such as Git, to track changes, collaborate, and manage the codebase effectively.

6. Documentation:

Documentation tools like Jupyter Notebooks, Markdown, or a README file should be maintained to explain the purpose of the code, installation steps, and usage instructions.

7. Testing Framework:

While the code does not explicitly include unit tests, incorporating a testing framework like pytest can be beneficial for future code enhancements and maintenance.

8. Model Deployment (Optional):

If you plan to deploy the trained model, consider additional tools or frameworks like Flask or FastAPI for creating a simple API. This allows integration into other applications or systems.

9. Dependency Management:

Utilize a package manager such as pip or conda to manage dependencies and ensure that the required libraries and versions are installed correctly.

10. Machine Learning Model Explanation (Optional):

If model interpretability is crucial, tools like SHAP (SHapley Additive exPlanations) can be employed to explain model predictions.

11. Containerization (Optional):

If you plan to deploy the code as a microservice, consider containerization tools like Docker for packaging the application and its dependencies.

12. Documentation Standards:

Adhere to documentation standards, such as PEP 257 for docstring conventions, to ensure code readability and ease of understanding.

CHAPTER 4

METHODOLOGY

1. Data Loading and Exploration:

Import necessary libraries: NumPy, Matplotlib, Pandas, Seaborn, and scikit-learn.
Load the Breast Cancer dataset from scikit-learn.
Print the dataset details and display the first and last few rows for initial exploration.
Check the dimensions (rows and columns) of the dataset.
Use descriptive statistics and visualizations to gain insights into the data.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import sklearn.datasets

# Load Breast Cancer dataset
breast_cancer_dataset = sklearn.datasets.load_breast_cancer()

# Display dataset details
print(breast_cancer_dataset)

# Load data into a Pandas DataFrame
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns=breast_cancer_dataset.feature_names)

# Display first and last few rows
print(data_frame.head())
print(data_frame.tail())

# Explore dataset dimensions and statistics
print(data_frame.shape)
print(data_frame.describe())
```

2. **Data Preprocessing:**

Separate features (X) and target variable (y).
Split the dataset into training and testing sets using train_test_split from scikit-learn.

```
from sklearn.model_selection import train_test_split

# Separate features and target variable
x = data_frame.drop(columns="label", axis=1)
y = data_frame["label"]

# Split dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
```

3. **Model Training:**

Create a Logistic Regression model using LogisticRegression from scikit-learn.
Train the model on the training dataset using the fit method.

```
from sklearn.linear_model import LogisticRegression

# Create Logistic Regression model
model = LogisticRegression()

# Train the model
model.fit(x_train, y_train)
```

4. **Model Evaluation:**

Evaluate the accuracy of the model on both training and testing datasets.
Print and analyze the accuracy scores.

```
from sklearn.metrics import accuracy_score # Accuracy on training data
x_train_prediction = model.predict(x_train) training_data_accuracy =
```

accuracy_score(y_train, x_train_prediction) print("Accuracy on training data =", training_data_accuracy) # Accuracy on test data x_test_prediction = model.predict(x_test) test_data_accuracy = accuracy_score(y_test, x_test_prediction) print("Accuracy on test data =", test_data_accuracy)

5. **Predictive System:**

Create an example input data point for prediction.
Reshape and predict using the trained Logistic Regression model.
Output whether the breast cancer is predicted as Malignant or Benign.
Example input data for prediction input_data = (13.54, 14.36, 87.46, 566.3, 0.09779, 0.08129, 0.06664, 0.04781, 0.1885, 0.05766, 0.2699, 0.7886, 2.058, 23.56, 0.008462, 0.0146, 0.02387, 0.01315, 0.0198, 0.0023, 15.11, 19.26, 99.7, 711.2, 0.144, 0.1773, 0.239, 0.1288, 0.2977, 0.07259) # Convert input data to a NumPy array and reshape input_data_as_numpy_array = np.asarray(input_data) input_data_reshaped = input_data_as_numpy_array.reshape(1, -1) # Predict using the trained model prediction = model.predict(input_data_reshaped) # Output the prediction result if prediction[0] == 0: print("The Breast Cancer is Malignant") else: print("The Breast Cancer is Benign")

6. **Documentation and Reporting:**

Maintain clear documentation using docstrings and comments to explain the purpose of each section of code.
Report key findings, accuracy metrics, and predictions in a readable and organized format.

CHAPTER 5

SYSTEM DESIGN

1. High-Level System Architecture:

The system is designed as a Python-based application, utilizing various libraries for data analysis and machine learning.

2. Modules:

Data Loading and Exploration:

Utilizes NumPy, Pandas, Matplotlib, and Seaborn for loading and exploring the Breast Cancer dataset.

Data Preprocessing:

Prepares the dataset by separating features and target variables, and splitting into training and testing sets.

Model Training and Evaluation:

Employs scikit-learn's Logistic Regression for creating and training the machine learning model.

Evaluates the model's accuracy on both training and testing datasets.

Predictive System:

Demonstrates the real-world application of the trained model with an example input for breast cancer prediction.

3. Data Flow:

Input: Breast Cancer dataset from scikit-learn.

Processing Steps:

Data exploration, preprocessing, and model training.

Model evaluation on training and testing datasets.

Predictive system demonstration using new input data.

Output: Model predictions and accuracy metrics.

4. Model Architecture:

Utilizes Logistic Regression as the primary machine learning algorithm for binary classification.

Input features include various attributes related to breast tumor characteristics.

5. Predictive System:

Input:

Example input data point representing tumor characteristics.

Processing Steps:

Reshape and predict using the trained Logistic Regression model.

Output whether the breast cancer is predicted as Malignant or Benign.

Output:

Prediction result indicating the nature of breast cancer.

6. Documentation and Reporting:

Documentation:

Utilizes docstrings and comments for explaining code sections.

Reporting:

Summarizes key findings, accuracy metrics, and predictions in a clear and organized format.

7. Potential Enhancements:

Feature engineering, hyperparameter tuning, and exploring other machine learning algorithms are highlighted as potential areas for future enhancement.

Consideration for model interpretability and deployment as a service is outlined for further development.

8. Security Considerations:

If handling sensitive data, security measures should be implemented. Encryption and access controls are examples of security considerations.

9. Error Handling:

The system includes mechanisms for error handling, especially during data loading, preprocessing, and model training.

10. Testing Framework (Optional):

Although not explicitly included in the provided code, incorporating a testing framework like pytest for unit tests is recommended for code reliability.

11. Version Control:

The use of Git for version control is suggested to track changes and facilitate collaboration.

12. Code Optimization (Optional):

Optimizing code for efficiency, especially considering computational resources, is noted as an optional consideration.

This system design outlines the architecture, modules, data flow, and key processes involved in the Breast Cancer Prediction code. It serves as a roadmap for understanding the structure and functionality of the codebase.

CHAPTER 6
IMPLEMENTATION AND RESULTS

I. Open your preferred Python development environment or a Jupyter Notebook.
II. Copy and paste the provided code into the editor.
III. Make sure you have the required libraries installed with the help of cmd. You can install them using the following command if you haven't already:

pip install numpy matplotlib pandas seaborn scikit-learn

Run the code blocks sequentially.
Here is the complete code for your convenience:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Loading the datafrom sklearn
breast_cancer_dataset = sklearn.datasets.load_breast_cancer()

print (breast_cancer_dataset)
# Ladin data to a data frame
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns =
  breast_cancer_dataset.feature_names)
# Print the first five rows
data_frame.head()
# Adding the target column to the data frame
data_frame["label"] = breast_cancer_dataset.target

# print the last 5 rows of dataset
```

```
data_frame.tail()

# number of rows and column of dataset
data_frame.shape

# getting some info of data
data_frame.info()

# checking missng values
data_frame.isnull().sum()

# Statistical measure about dataset
data_frame.describe()

# checking the distribution of target values

data_frame["label"].value_counts()

data_frame.groupby("label").mean()

x = data_frame.drop(columns="label", axis=1)
y = data_frame["label"]

print(x)
print(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
  random_state=2)

print(x.shape, x_train.shape, x_test.shape)

model = LogisticRegression()

model.fit(x_train, y_train)

# accuracy of training data
```

```
x_train_prediction = model.predict(x_train)
training_data_accuracy = accuracy_score(y_train, x_train_prediction)

print("Accuracy on training data = ", training_data_accuracy)

# accuracy of test data
x_test_prediction = model.predict(x_test)
test_data_accuracy = accuracy_score(y_test, x_test_prediction)

print("Accu", test_data_accuracy)

# building a predictive system
input_data                                                                    =
   (13.54,14.36,87.46,566.3,0.09779,0.08129,0.06664,0.04781,0.1885,0.
   05766,0.2699,0.7886,2.058,23.56,0.008462,0.0146,0.02387,0.01315,0
   .0198,0.0023,15.11,19.26,99.7,711.2,0.144,0.1773,0.239,0.1288,0.297
   7,0.07259)

# change input da into numpy array

input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array as we are predicting for one datapoint

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if(prediction[0] == 0):
   print("The Breast Cancer is Malignant")
else:
   print("The Breast Cancer is Benign")
```

Results

1. Data Exploration:

You'll see details of the Breast Cancer dataset printed, including information about features and target variables.

The first and last few rows of the dataset will be displayed.

2. Data Preprocessing:

The dataset will be separated into features (X) and the target variable (y).

The dataset will be split into training and testing sets.

3. Model Training:

A Logistic Regression model will be created and trained on the training dataset.

4. Model Evaluation:

Accuracy on the training and testing datasets will be printed.

5. Predictive System:

An example input data point will be provided and used to predict whether the breast cancer is malignant or benign.

The prediction result will be printed.

Example Output:
Accuracy on training data = 0.9472527472527472
Accuracy on test data = 0.9210526315789473
The Breast Cancer is Malignant/Benign

.

CHAPTER 7

CONCLUSION

The implementation of the Breast Cancer Prediction code using Logistic Regression provides valuable insights into the application of machine learning for early detection and diagnosis of breast cancer. Here are the key conclusions based on the results and methodology:

1. **Data Exploration:**

The dataset from scikit-learn contains information about various features related to breast tumor characteristics.

Initial exploration, including statistical measures and visualizations, provides a foundation for understanding the dataset.

2. **Model Training and Evaluation:**

Logistic Regression, chosen as the machine learning algorithm, demonstrates its effectiveness in binary classification for breast cancer prediction.

The model achieves high accuracy on both the training and testing datasets, indicating its ability to generalize well to new, unseen data.

3. **Predictive System:**

The trained model is capable of making predictions on new input data points.

An example input data point is provided, and the model predicts whether the breast cancer is Malignant or Benign.

4. **Accuracy Assessment:**

The accuracy scores on both the training and testing datasets are reasonably high, suggesting that the model performs well in distinguishing between malignant and benign tumors.

5. **Real-World Applicability:**

The model, when integrated into a predictive system, showcases its potential real-world application in assisting medical professionals with early diagnosis.

6. **Future Directions:**

Further enhancements can be explored, including feature engineering, hyperparameter tuning, and the investigation of other machine learning algorithms to potentially improve model performance.

Consideration for model interpretability and deployment as a service could enhance its practical utility.

7. **Security and Reliability:**

Security considerations, such as encryption and access controls, should be prioritized if handling sensitive patient data.

The implementation includes error handling mechanisms to ensure the reliability of the code during data processing and model training.

In conclusion, the implemented code provides a solid foundation for breast cancer prediction. The high accuracy achieved on testing data and the successful application of the model to new input data underscore its potential contribution to the field of medical diagnostics. Further research and improvements can build upon this foundation to create more sophisticated and accurate predictive models for breast cancer diagnosis.