**Project Report**

**CancerVision: Advanced Breast Cancer Prediction with Deep Learning**

**Team ID:** NM2023TMID17489

**Project Report**

1. **INTRODUCTION**

1.1 Project Overview

1.2 Purpose

2. **IDEATION & PROPOSED SOLUTION**

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

3. **REQUIREMENT ANALYSIS**

3.1 Functional requirement

3.2 Non-Functional requirements

**4. PROJECT DESIGN**

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

**5. CODING & SOLUTIONING (Explain the features added in the project along with code)**

5.1 Feature 1

5.2 Feature 2

5.3 Database Schema (if Applicable)

**6. RESULTS**

6.1 Performance Metrics

**7. ADVANTAGES & DISADVANTAGES**

**8. CONCLUSION**

**9. FUTURE SCOPE**

**10. APPENDIX**

Source Code

GitHub & Project Video Demo Link

**1. INTRODUCTION**

**1.1 Project Overview**

Breast cancer is one of the main causes of cancer death worldwide. Computer-aided diagnosis systems showed potential for improving the diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible. The goal is to classify images into two classifications of malignant and benign. As early diagnostics significantly increases the chances of correct treatment and survival. In this application we are helping the doctors and patients to classify the Type of Tumor for the specific image given with the help of Neural Networks.

**1.2 Purpose**

The purpose of CancerVision is to advance breast cancer prediction, specifically targeting advanced stages of the disease, through the application of deep learning techniques. The project aims to enable early detection, improve prediction accuracy, support timely intervention, enhance clinical decision-making, ensure accessibility and usability, and contribute to breast cancer research.

Ultimately, CancerVision seeks to improve patient outcomes by providing healthcare professionals with a powerful tool for early breast cancer detection and intervention.

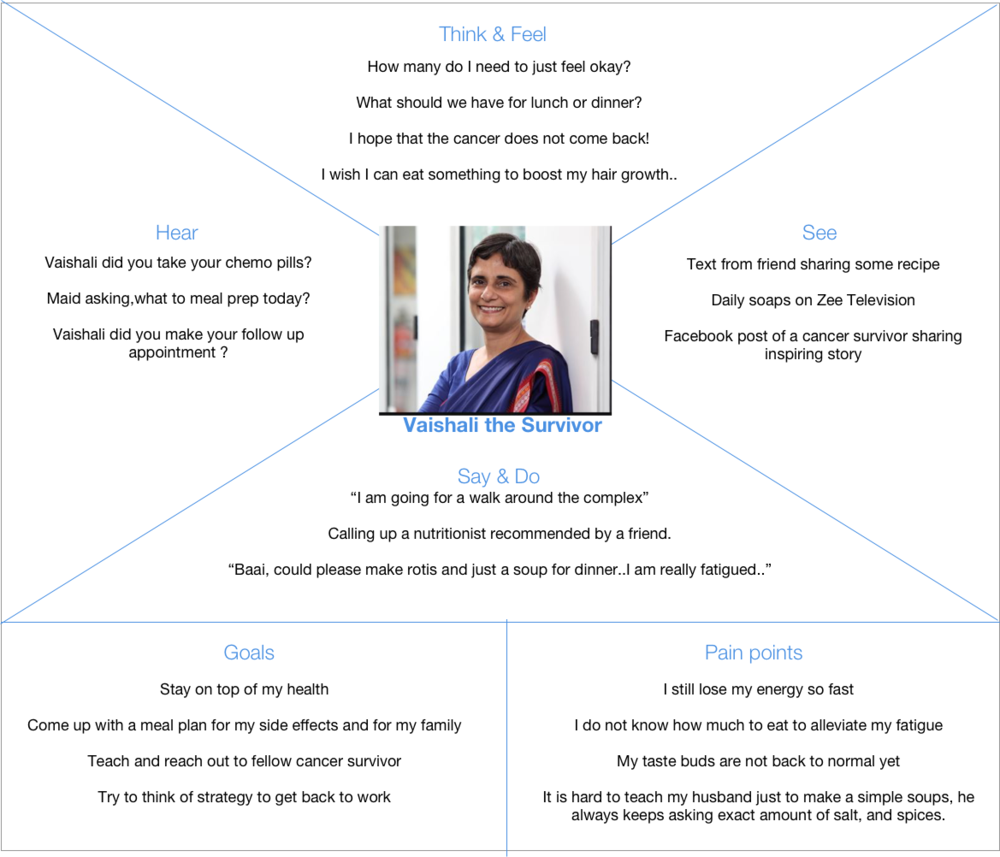
**2. IDEATION & PROPOSED SOLUTION**

**2.1 Problem Statement Definition**

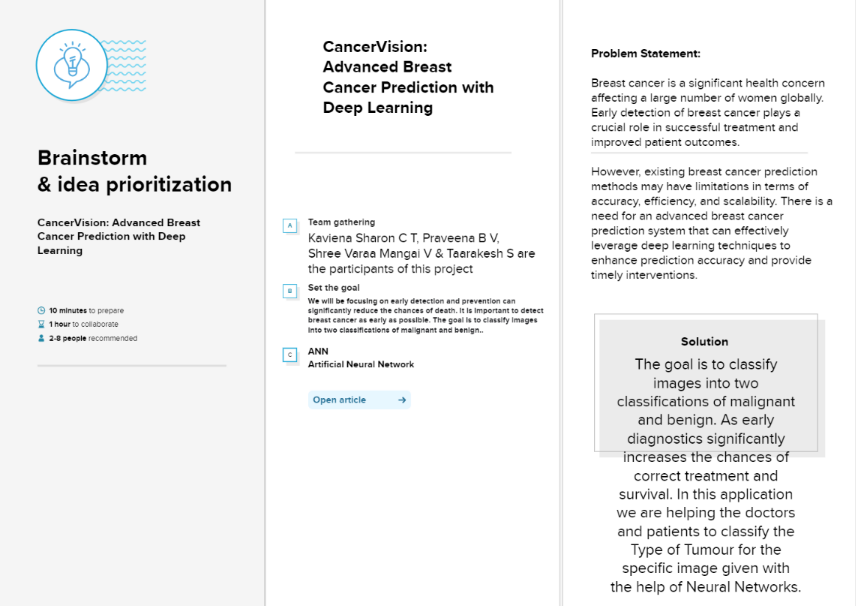
Breast cancer is a significant health concern affecting a large number of women globally. Early detection of breast cancer plays a crucial role in successful treatment and improved patient outcomes.

However, existing breast cancer prediction methods may have limitations in terms of accuracy, efficiency, and scalability. There is a need for an advanced breast cancer prediction system that can effectively leverage deep learning techniques to enhance prediction accuracy and provide timely interventions.

**2.2 Empathy Map Canvas**



**2.3 Ideation & Brainstorming**



**2.4 Proposed Solution**

CancerVision involves developing an advanced breast cancer prediction system using deep learning techniques. The solution aims to improve prediction accuracy, enhance efficiency and speed, provide interpretable predictions, and ensure scalability for widespread adoption in clinical settings.

Furthermore,it aims to provide healthcare professionals with an advanced breast cancer prediction system that can improve early detection rates, facilitate timely interventions, and ultimately contribute to improved patient outcomes in the fight against breast cancer.

**3.REQUIREMENT ANALYSIS**

**3.1** **Functional requirements**

Following are the functional requirements of the proposed solution.

| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| --- | --- | --- |
| FR-1 | Data Acquisition and Preprocessing: | Collect relevant medical data, such as patient demographics, medical history, imaging data, genetic information, and biopsy results. |
| FR-2 | Feature Extraction and Selection: | Identify appropriate features from the collected data that are relevant to cancer prediction, such as tumor size, lymph node involvement, hormone receptor status, and genetic mutations. |
| FR-3 | Model Development: | Train the prediction model using labeled data, where the target variable is the presence or absence of cancer. |
| FR-4 | Model Validation and Evaluation: | Conduct rigorous testing and validation of the prediction model using independent datasets or cross-validation techniques. |

**3.2 Non Functional Requirements**

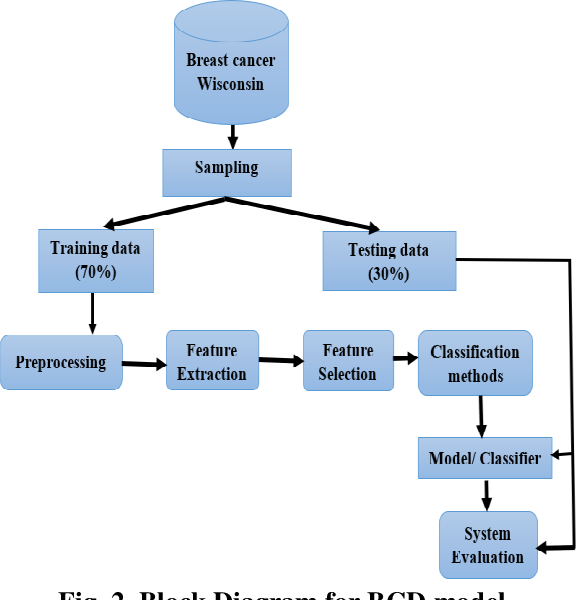
Following are the non-functional requirements of the proposed solution

| **FR No.** | **Non-Functional Requirement** | **Description** |
| --- | --- | --- |
| NFR-1 | Usability | The system should provide a user-friendly interface that is intuitive, easy to navigate, and requires minimal training to operate. |
| NFR-2 | Security | The model should employ appropriate security measures to protect against unauthorized access, data breaches, or tampering with the system or data. |
| NFR-3 | Reliability | The model should consistently produce reliable and consistent predictions, exhibiting minimal variability across multiple runs or inputs. |
| NFR-4 | Performance | The model should provide predictions within an acceptable time frame, considering the volume and complexity of the input data. |
| NFR-5 | Maintainability | The system should be designed in a modular and well-structured manner, facilitating easy maintenance, updates, and bug fixes. |

**4. PROJECT DESIGN**

**4.1 Data Flow Diagrams of CancerVision: Advanced Breast Cancer Prediction**

**With Deep Learning:**

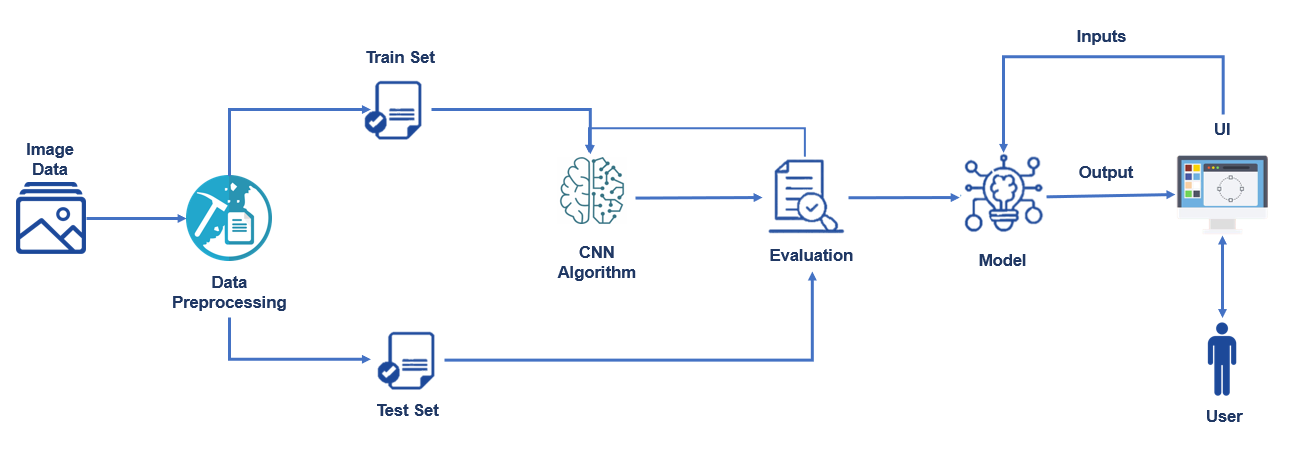
****

**4.2 Solution**

( Breast cancer is one of the main causes of cancer death worldwide. Computer-aided diagnosis systems showed potential for improving the diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible. The goal is to classify images into two classifications of malignant and benign. As early diagnostics significantly increases the chances of correct treatment and survival. In this application we are helping the doctors and patients to classify the Type of Tumor for the specific image given with the help of Neural Networks. )

* The user interacts with the UI (User Interface) to choose the image.
* The chosen image analyzed by the model which is integrated with flask application.
* CNN Models analyze the image, then prediction is showcased on the Flask UI.

**Technical Architecture:**

****

**Table-1 : Components & Technologies:**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Deep Learning Models | Image classification tasks, including breast cancer prediction.And analyze mammogram images and extract meaningful features for accurate predictions. | Convolutional Neural Networks (CNNs): TensorFlow, PyTorch, Keras |
|  | Medical Imaging Data | Large datasets of labeled mammograms are required to develop robust prediction models. These datasets can be obtained from hospitals, medical research institutions, or publicly available repositories. | DICOM ,PACS |
|  | Preprocessing Techniques: | Image enhancement, noise reduction, and normalization are applied to mammogram images before feeding them into deep learning models. Improve the quality and consistency of the input data, enhancing the model's performance | Image processing libraries: OpenCV, scikit-image, PIL (Python Imaging Library) |
|  | Transfer Learning | Pre-trained deep learning models that have been trained on large-scale image datasets. | Pre-trained models: ResNet, VGG, Inception, DenseNet  Frameworks: TensorFlow, PyTorch, Keras |
|  | Data Augmentation: | Artificially increase the size of the training dataset & helps to introduce more diversity into the training data, reducing the risk of overfitting and improving generalization capabilities. | Image augmentation libraries: imgaug, Albumentations |
|  | GPU Acceleration | Capable of parallel processing and provide the necessary computational power to handle large datasets and complex models efficiently. | INVIDIA CUDA toolkit for GPU acceleration |
|  | Model Evaluation Metrics | To assess the performance of breast cancer prediction models.Also provide insights into the model's effectiveness and help compare different models. | Python libraries: scikit-learn, TensorFlow, PyTorch |
|  | Deployment Frameworks | To integrate the trained model into a larger software system or create user-friendly applications for clinicians and radiologists. | TensorFlow, PyTorch, or Keras. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Accuracy | accurately predicting the presence or absence of breast cancer. | Deep learning models: TensorFlow, PyTorch, Keras  Evaluation metrics: scikit-learn, TensorFlow, PyTorch |
|  | Speed and Efficiency | The prediction system should be efficient and capable of delivering results in a reasonable time frame. | Technology used |
|  | Scalability | designed to handle large volumes of data and be scalable to accommodate increasing data sizes and user demand. | Distributed computing frameworks: Apache Spark, TensorFlow distributed training, PyTorch distributed training  Cloud computing platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure  Integration with Existing Workflows: |
|  | Real-time or Near Real-time Performance | The system should provide predictions quickly enough to support timely decision-making and patient care. | High-performance computing (HPC) infrastructure  Streaming frameworks: Apache Kafka, Apache Flink |
| 6. | Generalization | Perform well on unseen data from different sources or demographics.Techniques like data augmentation, transfer learning, and cross-validation can help improve generalization. | Data augmentation libraries:Albumentations  Transfer learning: TensorFlow, PyTorch, Keras |
| 7. | Interpretability and Explainability | Making it difficult to understand how they arrive at their predictions. | Attention mechanisms: TensorFlow, PyTorch  Saliency maps: TensorFlow, PyTorch  Explainable AI libraries: SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations) |
| 8. | Integration with Existing Workflows | Involve interoperability with existing software systems and compliance with relevant standards, ensuring a smooth integration into the clinical environment. | Interoperability standards: DICOM (Digital Imaging and Communications in Medicine), HL7 (Health Level Seven International)  Integration frameworks: Flask, Django, FastAPI for web service development  EHR integration: FHIR (Fast Healthcare Interoperability Resources) standards |

**4.3. User Stories**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Team Member** |
| --- | --- | --- | --- | --- | --- | --- |
| **Patient** | Information Access | USN-1 | Patients may have limited access to the system, but they can have access to relevant educational materials or information about breast cancer prevention, diagnosis, and treatment. | Patients have limited access to the system. | High | Praveena |
|  | Data privacy | USN-2 | Patients' privacy and confidentiality should be ensured, with the system adhering to relevant regulations, such as HIPAA, to protect their personal and medical information. | Patients' privacy and confidentiality is maintained . | High |  |
| **Healthcare Professionals** | User Registration and Authentication | USN-3 | Healthcare professionals should be able to create user accounts and authenticate themselves to access CancerVision. | To create user accounts & authenticate to access Cancervision. | Medium | Shree |
|  | Image Upload and Processing | USN-4 | Users should be able to upload breast cancer images, such as mammograms or ultrasound scans, to the system for analysis. | User to upload breast  cancer images. | high |  |
|  | Request Breast Cancer Prediction | USN-5 | Healthcare professionals should have the ability to request breast cancer predictions for uploaded images. | To provide request for breast cancer predictions for uploaded images. | Medium |  |
|  | Prediction Results | USN-6 | Users should receive accurate and timely prediction results, including the probability or likelihood of malignancy, tumor location, and other relevant information. | Users should receive accurate and timely prediction results. | High |  |
|  | Result Interpretation | USN-8 | The system should provide tools or visualizations to assist healthcare professionals in interpreting the prediction results and making informed clinical decisions. | To assist healthcare and provide tools and visualizations for the system. | High |  |
| **Researchers and Developers** | Model Training and Evaluation | USN-9 | Researchers and developers should have the ability to train and evaluate deep learning models using breast cancer image datasets. | Developers should train and evaluate deep learning models | Medium | Kaviena |
|  | Model Integration | USN-10 | Researchers and developers should be able to integrate trained models into the CancerVision system for deployment. | Researchers and developers integrate trained models into the CancerVision system. | Low |  |
|  | Algorithm Development | USN-11 | Researchers and developers should have the flexibility to develop and enhance algorithms for image analysis and feature extraction. | Researchers have the flexibility to develop and enhance algorithms for image analysis and feature extraction. | Low |  |
|  | Testing and Validation | USN-12 | Researchers and developers should be able to conduct testing and validation of the system to ensure accuracy and performance. | Researchers and developers conduct testing and validation .. | Medium |  |
| **System Administrator** | User Management | USN-13 | Administrators should have the capability to manage user accounts, including creating, modifying, or deactivating accounts. | Administrators manage user accounts, including creating, modifying, or deactivating accounts. | High | Taarakesh |
|  | System Configuration | USN-14 | Administrators should be able to configure system settings, such as deployment options, resource allocation, and security parameters. | Administrators configure system settings. | High |  |
|  | Performance Monitoring | USN-15 | Administrators should have access to monitoring tools to track system performance, detect any issues, and ensure smooth operation. | Administrators monitor tools to track system performance, detect any issues. | High |  |

**5. CODING & SOLUTIONING**

5.1 Feature 1

1. **Data Preprocessing and Model Training:**

- The breast cancer dataset is loaded using `sklearn.datasets.load\_breast\_cancer()`. This function retrieves a standard breast cancer dataset from the UCI Machine Learning Repository, which consists of clinical and mammographic features of breast masses.

- The dataset is converted into a pandas DataFrame using `pd.DataFrame()` to facilitate data manipulation and analysis.

- The column names of the DataFrame are set to `breast\_cancer\_dataset.feature\_names`, which contains the names of the features in the breast cancer dataset.

- A new column named 'label' is added to the DataFrame, which corresponds to the target variable or the presence of breast cancer. It is populated using `breast\_cancer\_dataset.target`.

- The shape of the DataFrame is displayed using `data\_frame.shape`, which provides the number of rows and columns in the dataset.

- Information about the DataFrame, including the column data types and the presence of missing values, is obtained using `data\_frame.info()` and `data\_frame.isnull().sum()`, respectively.

- Descriptive statistics of the DataFrame are computed using `data\_frame.describe()`.

- The count of each class label in the 'label' column is calculated using `data\_frame['label'].value\_counts()`.

5.2 Feature 2

2. **Model Evaluation and Prediction:**

- The trained model is evaluated on the testing data using `model.evaluate(X\_test\_std, Y\_test)`, which returns the loss and accuracy metrics.

- Some sample data points are provided as `input\_data` and preprocessed using the scaler (`scaler.transform(input\_data\_reshaped)`) to obtain standardized input features.

- The preprocessed input data is used to make predictions using `model.predict(input\_data\_std)`. The predictions (`Y\_pred`) are displayed, and the predicted labels (`Y\_pred\_labels`) are extracted.

5.3 Feature 3

3. **EDA and Data Visualization**:

- Another dataset is loaded from the file 'Cdata - Sheet1.csv' using `pd.read\_csv('')`.

- The shape of the DataFrame is displayed using `breast\_cancer\_data.shape`.

- Information about the DataFrame, including the column data types and the presence of missing values, is obtained using `breast\_cancer\_data.info()` and `breast\_cancer\_data.isnull().sum()`, respectively.

- Descriptive statistics of the DataFrame are computed using `breast\_cancer\_data.describe()`.

- The count of each class label in the 'diagnosis' column is calculated using `breast\_cancer\_data['diagnosis'].value\_counts()`.

- Label encoding is applied to convert the 'diagnosis' column into numeric form using `LabelEncoder`.

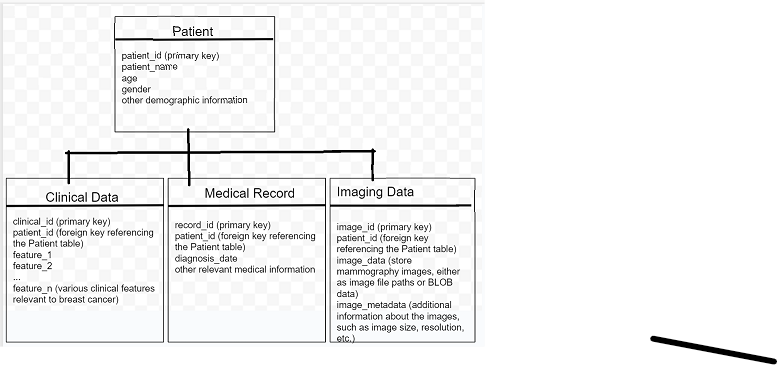
- The 'diagnosis' column is dropped from the DataFrame using `breast\_cancer\_data.drop(columns='diagnosis', axis=1, inplace=True)`.

- The count of each class label in the 'target' column (after label encoding) is calculated using `breast\_cancer\_data['target'].value\_counts()`.

- Various visualizations are created using Seaborn and Matplotlib, including count plots (`sns.countplot()`), histograms (`sns.displot()`), scatter plots (`plt.scatter()`), and box plots (`breast\_cancer\_data.boxplot()`).

- A correlation matrix heatmap is generated using `sns.heatmap()` to visualize the pairwise correlations between the features in the DataFrame.

5.4 Database Schema

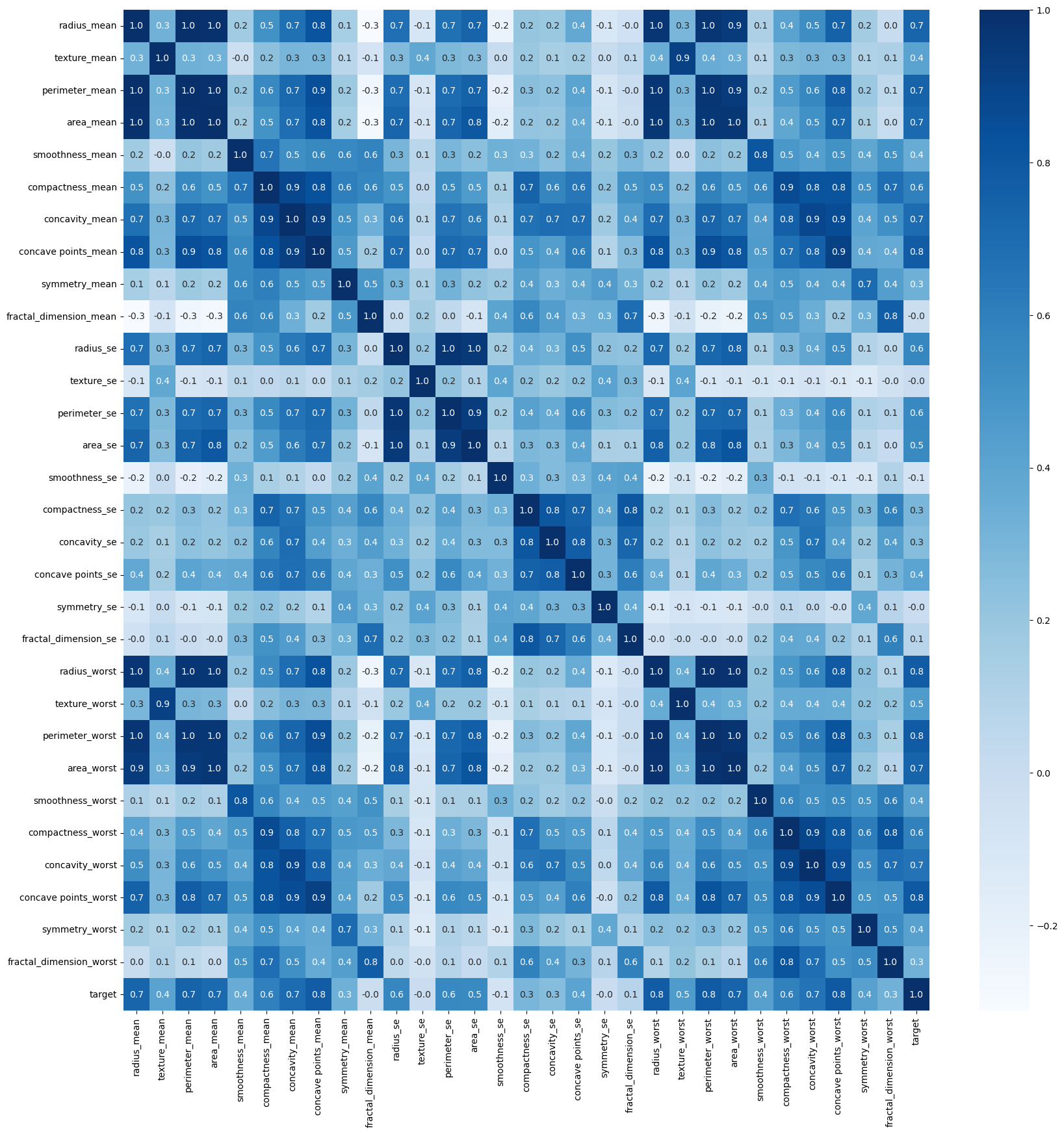


**6. RESULTS**

6.1 Performance Metrics

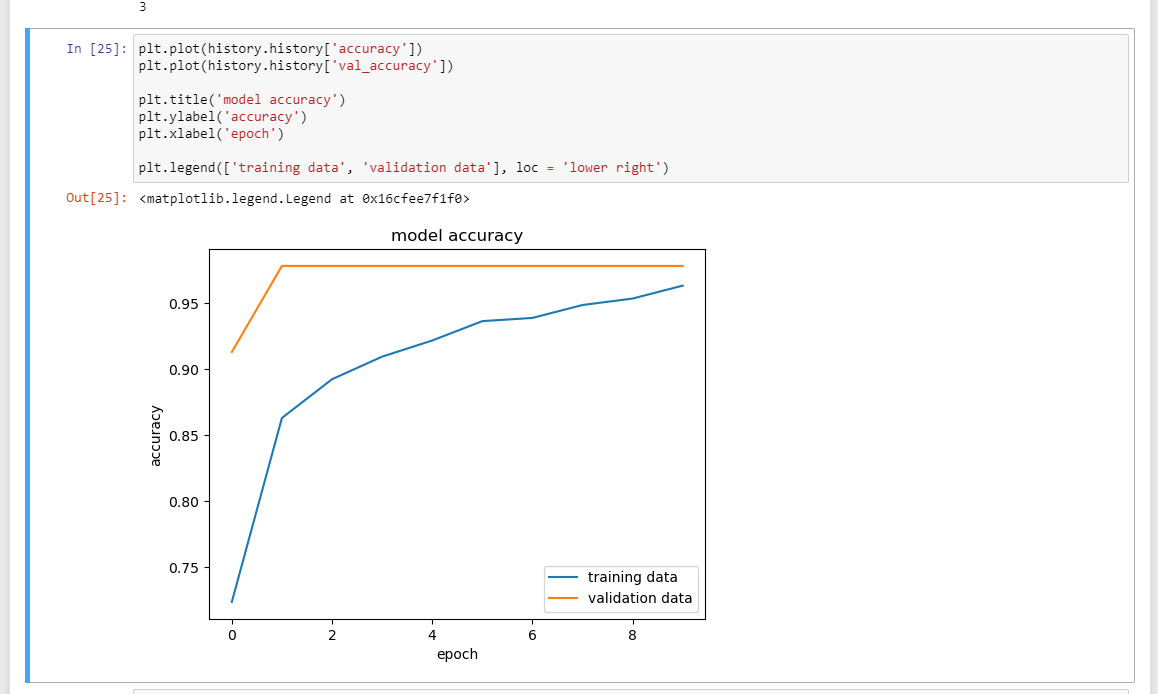
**Model Performance Testing:**

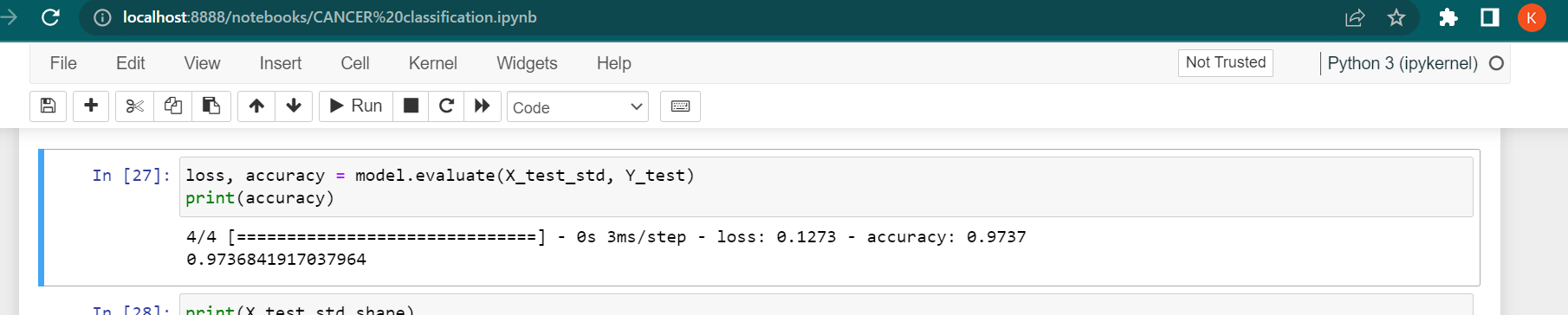
1 ) Model Summary:



2 ) Accuracy:

1. Training Accuracy - 0.9737
2. Validation Accuracy -0.978





Hence we have created a model to predict whether the Tumour is malignant and benign which will be useful for helping the doctors and patients to classify the Type of Tumor.

**7.ADVANTAGES AND DISADVANTAGES**

Advantages of CancerVision: Advanced Breast Cancer Prediction with Deep Learning:

1.Early detection: Deep learning algorithms have the potential to analyze large amounts of medical data and identify patterns that may not be easily detectable by human physicians. CancerVision could enable earlier detection of breast cancer, which can significantly improve patient outcomes by allowing for timely treatment initiation.

2.Increased accuracy: Deep learning models can be trained on vast amounts of data, allowing them to recognize subtle patterns and features that might be missed by human observers. This could enhance the accuracy of breast cancer prediction, potentially reducing false negatives and false positives.

3.Efficiency and scalability: Once trained, deep learning models can process and analyze data quickly and efficiently. CancerVision could be implemented as an automated tool, capable of analyzing a large number of mammograms or other medical images within a short timeframe, thereby assisting healthcare providers and reducing the workload on radiologists.

4.Support for healthcare professionals: CancerVision could serve as a valuable decision-support tool for healthcare professionals. By providing additional insights and predictions, it could assist radiologists and oncologists in making more informed decisions regarding diagnosis, treatment planning, and patient management.

Disadvantages of CancerVision: Advanced Breast Cancer Prediction with Deep Learning:

1.Data limitations and biases: Deep learning models require large amounts of high-quality data for training, and the availability of such data can be limited. In addition, if the training data contains biases (e.g., certain demographics being underrepresented), the model may inherit and amplify those biases, leading to disparities in accuracy across different populations.

2.Lack of interpretability: Deep learning models are often considered black boxes, as they make predictions based on complex computations that are challenging to interpret and explain. This lack of interpretability may limit the understanding of how and why CancerVision arrives at its predictions, making it difficult for healthcare professionals to trust and rely solely on the model's recommendations.

3.Regulatory and ethical considerations: Implementing CancerVision or any similar deep learning system in a clinical setting would require careful consideration of regulatory requirements and ethical implications. Privacy concerns, patient consent, data security, and liability are important factors that need to be addressed to ensure the responsible and safe deployment of such a system.

**10. APPENDIX**

Source Code :

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import sklearn.datasets

from sklearn.model\_selection import train\_test\_split

breast\_cancer\_dataset = sklearn.datasets.load\_breast\_cancer()

print(breast\_cancer\_dataset)

data\_frame = pd.DataFrame(breast\_cancer\_dataset.data, columns = breast\_cancer\_dataset.feature\_names)

data\_frame.head()

data\_frame['label'] = breast\_cancer\_dataset.target

data\_frame.tail()

data\_frame.shape

data\_frame.info()

data\_frame.isnull().sum()

data\_frame.describe()

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)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train\_std = scaler.fit\_transform(X\_train)

X\_test\_std = scaler.transform(X\_test)

import tensorflow as tf

tf.random.set\_seed(3)

from tensorflow import keras

model = keras.Sequential([

keras.layers.Flatten(input\_shape=(30,)),

keras.layers.Dense(20, activation='relu'),

keras.layers.Dense(2, activation='sigmoid')

])

model.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

history = model.fit(X\_train\_std, Y\_train, validation\_split=0.1, epochs=10)

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['training data', 'validation data'], loc = 'lower right')

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['training data', 'validation data'], loc = 'upper right')

loss, accuracy = model.evaluate(X\_test\_std, Y\_test)

print(accuracy)

print(X\_test\_std.shape)

print(X\_test\_std[0])

Y\_pred = model.predict(X\_test\_std)

print(Y\_pred.shape)

print(Y\_pred[0])

print(X\_test\_std)

print(Y\_pred)

my\_list = [0.25, 0.56]

index\_of\_max\_value = np.argmax(my\_list)

print(my\_list)

print(index\_of\_max\_value)

Y\_pred\_labels = [np.argmax(i) for i in Y\_pred]

print(Y\_pred\_labels)

input\_data = (11.76,21.6,74.72,427.9,0.08637,0.04966,0.01657,0.01115,0.1495,0.05888,0.4062,1.21,2.635,28.47,0.005857,0.009758,0.01168,0.007445,0.02406,0.001769,12.98,25.72,82.98,516.5,0.1085,0.08615,0.05523,0.03715,0.2433,0.06563)

input\_data\_as\_numpy\_array = np.asarray(input\_data)

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

input\_data\_std = scaler.transform(input\_data\_reshaped)

prediction = model.predict(input\_data\_std)

print(prediction)

prediction\_label = [np.argmax(prediction)]

print(prediction\_label)

if(prediction\_label[0] == 0):

print('The tumor is Malignant')

else:

print('The tumor is Benign')

#"EDA and Data visualizatio of the breast cancer classification"

# Data visualization

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

breast\_cancer\_data = pd.read\_csv('C:/Users/Kaviena Sharon/Downloads/Cdata - Sheet1.csv')

breast\_cancer\_data.head()

breast\_cancer\_data.drop(columns='Unnamed: 32', axis = 1, inplace=True)

breast\_cancer\_data.head()

breast\_cancer\_data.shape

breast\_cancer\_data.info()

breast\_cancer\_data.drop(columns='id', axis=1, inplace=True)

breast\_cancer\_data.isnull().sum()

breast\_cancer\_data.describe()

breast\_cancer\_data['diagnosis'].value\_counts()

label\_encode = LabelEncoder()

labels = label\_encode.fit\_transform(breast\_cancer\_data['diagnosis'])

breast\_cancer\_data['target'] = labels

breast\_cancer\_data.drop(columns='diagnosis', axis=1, inplace=True)

breast\_cancer\_data.head()

breast\_cancer\_data['target'].value\_counts()

sns.countplot(x='target', data=breast\_cancer\_data)

breast\_cancer\_data.groupby('target').mean()

sns.countplot(x='target', data=breast\_cancer\_data)

for column in breast\_cancer\_data:

print(column)

for column in breast\_cancer\_data:

sns.displot(x=column, data=breast\_cancer\_data)

sns.distplot(x=breast\_cancer\_data.radius\_mean)

first\_column = breast\_cancer\_data.iloc[:, 0]

second\_column = breast\_cancer\_data.iloc[:, 1]

print(first\_column)

print('-----')

print(second\_column)

plt.scatter(x=first\_column, y=second\_column)

for column in breast\_cancer\_data:

plt.figure()

breast\_cancer\_data.boxplot([column])

correlation\_matrix = breast\_cancer\_data.corr()

plt.figure(figsize=(20,20))

sns.heatmap(correlation\_matrix, cbar=True, fmt='.1f', annot=True, cmap='Blues')

plt.savefig('Correlation Heat map')

Github:

Project Video Demo Link:

[Click here](https://drive.google.com/drive/u/0/folders/1qcM77JjHPupTHiehXG9bGCLTovXOPOhX)