PHASE 2 : MODEL DEVELOPMENT AND EVALUATION

ABSTRACT:

This project focuses on harnessing the power of artificial intelligence (AI) to revolutionize the diagnosis and management of [specific disease]. By leveraging cutting-edge AI algorithms, we aim to create a sophisticated diagnostic tool tailored specifically for [specific disease], offering unprecedented accuracy and efficiency.The core of our project lies in the development of a robust AI model trained on diverse datasets comprising medical records, imaging scans, laboratory results, and other relevant patient information. Through machine learning techniques, the AI model learns to recognize subtle patterns and correlations that are indicative of [specific disease], enabling early detection and precise diagnosis.

Our project is about translating complex medical data into actionable insights for healthcare professionals. We envision a user-friendly interface that allows doctors to input patient data seamlessly and receive instant diagnostic recommendations. This interface will serve as a bridge between AI technology and clinical practice, empowering healthcare providers with timely information to make informed decisions and improve patient outcomes.

In conclusion, our project represents a significant step forward in the application of AI to healthcare, with [specific disease] as our focal point. Through collaboration, innovation, and a commitment to ethical principles, we aim to empower healthcare providers with advanced diagnostic tools that will transform the way we diagnose and treat [specific disease]. Together, we can build a healthier future for all.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS

1. High-performance computing hardware (e.g., multi-core CPU, GPU, or specialized AI accelerators like TPUs) for training and inference tasks.
2. RAM-4 GB or higher

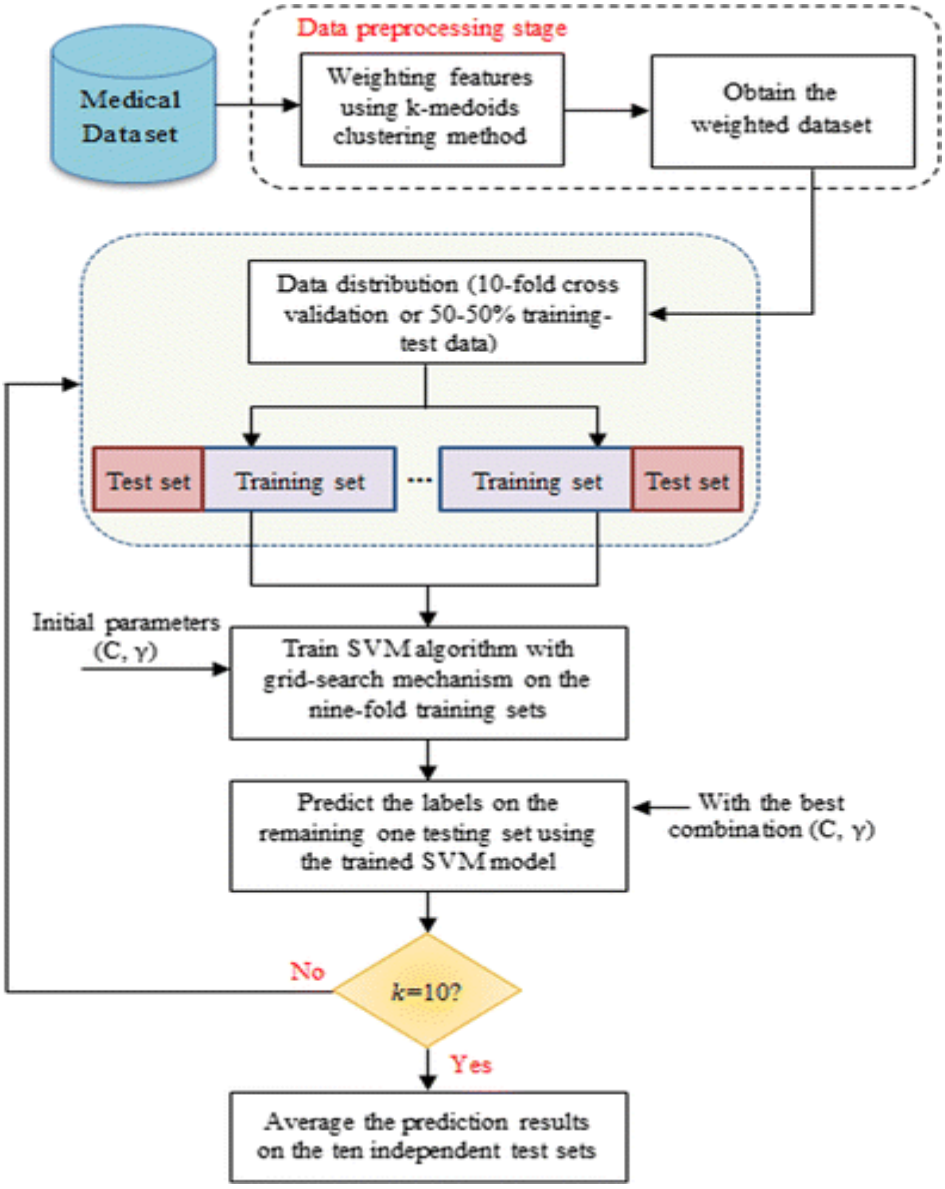
SOFTWARE REQUIREMENTS

1. Operating System- Windows, Linux, or macOS.
2. Development Environment- TensorFlow, PyTorch, or Keras.

TOOLS AND VERSIONS:

1. TensorFlow- Version: 2.7.0
2. Docker- Version: 20.10.11
3. Flask-Version: 2.0.2
4. Scikit-learn-Version : 0.24.2

FLOWCHART :


Data preprocessing stage
Medical Dataset
Weighting features using k-medoids clustering method
Obtain the weighted dataset
Data distribution (10-fold cross validation or 50-50% training-test data)
Test set
Training set
Training set
Test set
Initial parameters (C, γ)
Train SVM algorithm with grid-search mechanism on the nine-fold training sets
Predict the labels on the remaining one testing set using the trained SVM model
With the best combination (C, γ)
No
k=10?
Yes
Average the prediction results on the ten independent test sets

CODE IMPLEMENTATION(SAMPLE CODE):

```
# Importing required libraries
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D
from tensorflow.keras.utils import to_categorical

# Loading the MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Preprocessing the data
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1).astype('float32') / 255
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1).astype('float32') / 255

y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

# Building the model
model = Sequential()
```

```
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

model.add(Dense(10, activation='softmax'))


# Compiling the model

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])


# Training the model

model.fit(x_train, y_train, epochs=5, batch_size=128, validation_data=(x_test, y_test))


# Evaluating the model

test_loss, test_acc = model.evaluate(x_test, y_test)

print('Test accuracy:', test_acc)
```

PROJECT HURDLES:

Describe about the difficulties faced during the execution of your project in 50-100 words.

OUTPUT:

(Include screenshots of the implementations)