Code Documentation and Cluster Setup

# 1. Code Documentation

The source code implements a real-time blood pressure prediction system using Apache Spark. The main components of the code are:  
  
1. \*\*SparkSession\*\*: Initializes the Spark environment.  
2. \*\*Data Loading\*\*: Reads the blood pressure dataset in CSV format.  
3. \*\*Feature Engineering\*\*: Uses VectorAssembler to combine features (PTT, PIR, HR) into a single vector.  
4. \*\*Clustering\*\*: Applies K-Means clustering to identify patterns in the data.  
5. \*\*Regression Model\*\*: Trains a Random Forest Regressor to predict Systolic Blood Pressure (SBP).  
6. \*\*Evaluation\*\*: Uses Root Mean Square Error (RMSE) to evaluate model performance.

# 2. Cluster Setup Documentation

To run the blood pressure prediction system, a Spark and Kafka cluster must be set up as follows:  
  
### Apache Spark Setup:  
1. Install Spark and Java:  
 - `sudo apt install openjdk-11-jdk`  
 - Download and extract Spark from the official site.  
  
2. Set environment variables:  
 - `SPARK\_HOME=/path/to/spark`  
 - Add Spark to `PATH`.  
  
3. Start Spark standalone cluster:  
 - `sbin/start-master.sh`  
 - `sbin/start-worker.sh spark://<master-ip>:7077`  
  
### Apache Kafka Setup:  
1. Download and extract Kafka.  
2. Start Zookeeper:  
 - `bin/zookeeper-server-start.sh config/zookeeper.properties`  
3. Start Kafka broker:  
 - `bin/kafka-server-start.sh config/server.properties`  
4. Create topic:  
 - `bin/kafka-topics.sh --create --topic bp-stream --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1`  
  
### Integration:  
- Use Spark Structured Streaming to read from Kafka.  
- Stream preprocessed IoT data to Kafka.  
- Spark application consumes data for real-time prediction.