|  |  |  |
| --- | --- | --- |
|  | **Universidad de Costa Rica Programa de Posgrado en Ingeniería Eléctrica Maestría Académica en Ingeniería Eléctrica** | **A logo for a company  Description automatically generated** |
| **SP-2136 Programación Avanzada** | | |
| **Propuesta Formalización Proyecto** | | |
| **Alumno: Carlos Benavides Víquez** | | |

1. **Descripción general del problema que van a solucionar ya sea parcial o totalmente con el uso del paralelismo.**

El objetivo del proyecto es acelerar el proceso de predicción del riesgo de fractura ósea utilizando datos de DXA (absorciometría de rayos X de energía dual) y Senior Fitness Test (SFT) mediante el uso de un modelo de “Support Vector Machine” (SVM).

Este análisis implica el manejo de volúmenes potencialmente grandes y dinámicos de datos médicos relacionados con la densidad mineral ósea (BMD) y resultados de pruebas físicas que se correlacionan con el riesgo de fractura, como por ejemplo la fuerza muscular y el equilibrio.

El problema principal es que el entrenamiento del modelo SVM con estos datos es computacionalmente costoso y se requiere una solución que permita procesar y entrenar el modelo de manera eficiente para obtener resultados en tiempos razonables.

1. **Breve justificación del porque es necesaria la implementación del paralelismo en el problema planteado.**La implementación del paralelismo es sumamente conveniente debido a la naturaleza intensiva en tiempo de los cálculos involucrados en el preprocesamiento de los datos (escalado y selección de características), así como en el entrenamiento del modelo SVM.   
     
   Utilizar paralelismo permitirá dividir las tareas computacionales (como la normalización de los datos, la validación cruzada y la selección de características) entre múltiples núcleos de procesamiento, lo que acelerará significativamente el proceso de análisis. Esto es crucial para manejar datasets grandes y para realizar pruebas iterativas de hiperparámetros, lo que es típico en proyectos de aprendizaje automático.
2. **Especificar claramente cuál es la funcionalidad, código, algoritmos que van a paralelizar.**La primera parte del procesamiento de datos necesaria para el análisis del riesgo de fractura es la **normalización/estandarización de los datos DXA y SFT.**

Esto implica que cada característica (BMD, fuerza, agilidad, etc.) se escale simultáneamente en múltiples núcleos de procesamiento.

Por ejemplo, para cada paciente se pueden tener los siguientes datos

| **Metric** | **Value** |
| --- | --- |
| Hip BMD | 0.85 g/cm² |
| Spine BMD | 0.90 g/cm² |
| Femur BMD | 0.82 g/cm² |
| Lean Mass | 40 kg |
| Fat Mass | 25% |
| Bone Mass | 2.5 kg |
| Chair Stand Test | 12 stands |
| 8-Foot Up-and-Go | 6.5 seconds |
| Arm Curl Test | 15 curls |
| 6-Minute Walk Test | 500 meters |
| Chair Sit-and-Reach | -3 cm |
| Back Scratch Test | 2 cm gap |

Y luego del proceso de normalizacion/estandarizado, los datos lucirían de la siguiente manera:

| **Metric** | **Scaled Value** |
| --- | --- |
| Hip BMD | 1.5109662 |
| Spine BMD | 0.65465367 |
| Femur BMD | 0.50709255 |
| Lean Mass | 0.16903085 |
| Fat Mass | -0.4472136 |
| Bone Mass | 0.4472136 |
| Chair Stand Test | 0.4472136 |
| 8-Foot Up-and-Go | -0.19911699 |
| Arm Curl Test | 0.50709255 |
| 6-Minute Walk Test | 0.16903085 |
| Chair Sit-and-Reach | -1.34164079 |
| Back Scratch Test | -0.4472136 |

Este proceso de normalización/estandarización debe realizarse para **n** pacientes (potencialmente miles) y es posible que esta operacion se deba repetir varias veces con datos nuevos y probando diferentes tipos/mètodos de estandarizaciòn de datos.

1. **Posible bibliotecas o lenguajes de programación paralela a utilizar. Tomando en cuenta si esutilizando memoria compartida, memoria distribuida o ambos  
     
     
   Lenguaje**: El código se implementará en C, dada su eficiencia y capacidad para manejar operaciones a bajo nivel y aprovechar al máximo los recursos de hardware.  
   **Libreria:** OpenMP por su simplicidad y conveniencia para el manejo de este tipo de problemas que no requieren memoria destribuida, y que pueden resolverse eficientemente con memoria compartida