**Funcionamiento:**

Este proyecto se dividirá en 2 secciones:

* Dispositivo de laboratorio (Raspberry)
* Dispositivo de campo (ESP32)

**Dispositivo de laboratorio**

El dispositivo de laboratorio se encargará de crear toda una interfaz visual para poder interactuar con el dispositivo, comunicarse con el sensor, almacenar las muestras del sensor, procesar las muestras, clasificar, graficar, y crear modelos de clasificación óptimos, los cuales serán almacenarlos para su uso y utilizarlos para subiros a un servidor online.

Todo esto requiere un alto nivel computacional, por lo tanto, una Raspberry pi cumple los parámetros para realizar todas las tareas mencionadas anteriormente.

**Lenguaje**

Python será el lenguaje utilizado para este dispositivo de laboratorio gracias al amplio desarrollo que ha tenido últimamente por su comunidad presenta todas las cualidades para implantar el proyecto.

**Interfaz Visual**

La interfaz visual que se utilizara el framework Django, le cual esta basado en Python y permite crear una interfaz agradable y segura para la toma de datos. La programación de este framwork sería orientado a clase basada en vistas , se implantaron tres niveles de seguridad:

* Desarrollador
* Administrador
* Invitado

Esta interfaz visual será desplegada sobre apache2, para poder servirlo desde cualquier red externa.

**Comunicación con el sensor**

Para la comunicación con el sensor As7265x, es necesario la comunicación por el protocolo I2C, protocolo que la Raspberry pi soporta, la libraría smbus de Python permitirá la lectura de datos.

**Almacenamiento**

Django trae implementado por defecto sqlite3, base de datos que permite alancear de manera organizada las muestras

**Clasificación de datos**

Para la clasificación de datos existen múltiples métodos :

1. LDA (Análisis de Discriminante Lineal): El Análisis de Discriminante Lineal busca encontrar una combinación lineal de características que maximice la separación entre diferentes clases de datos. Proyecta los datos en un espacio de menor dimensión mientras maximiza la varianza entre las clases y minimiza la varianza dentro de cada clase.  
   Referencia: Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7(2), 179-188.
2. KNN (K-vecinos más cercanos): El algoritmo K-vecinos más cercanos clasifica un nuevo punto de datos según la clase mayoritaria de sus k vecinos más cercanos en el espacio de características. La distancia se calcula utilizando una medida de distancia, como la distancia euclidiana.  
   Referencia: Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.
3. Redes Neuronales: Las redes neuronales son modelos computacionales inspirados en el cerebro humano. Consisten en capas de nodos interconectados (neuronas) que procesan y transmiten información. Las redes neuronales pueden aprender a clasificar datos ajustando los pesos de las conexiones entre las neuronas durante un proceso de entrenamiento.  
   Referencia: Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536
4. Árboles de Decisión: Los árboles de decisión son estructuras de tipo árbol donde cada nodo interno representa una característica o atributo, y las ramas del árbol representan las posibles opciones o valores para esa característica. Los árboles de decisión se construyen dividiendo los datos en función de las características más relevantes y se utilizan para clasificar nuevos ejemplos siguiendo el camino desde la raíz hasta las hojas del árbol.  
   Referencia: Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.
5. SVM (Support Vector Machines): Las Máquinas de Vectores de Soporte son modelos de aprendizaje automático que buscan encontrar un hiperplano óptimo que separe las diferentes clases de datos en un espacio de características. El objetivo es maximizar el margen entre las instancias de diferentes clases más cercanas al hiperplano de decisión. SVM también puede utilizar kernels para manejar datos no linealmente separables.  
   Referencia: Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.
6. Naive Bayes (Clasificador Bayesiano Ingenuo): El clasificador bayesiano ingenuo se basa en el teorema de Bayes y asume independencia condicional entre las características. Calcula la probabilidad de que un ejemplo pertenezca a una clase determinada utilizando la probabilidad de la clase y la probabilidad de cada característica dado un valor de clase. A pesar de su suposición simplificada, este método puede ser efectivo en muchos casos.  
   Referencia: Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29(2-3), 103-130.
7. Random Forest (Bosques Aleatorios): Los bosques aleatorios son un conjunto de árboles de decisión combinados. Cada árbol se entrena con una muestra aleatoria del conjunto de datos y produce una clasificación. La clasificación final se obtiene por votación o promediando las predicciones de los árboles individuales. Los bosques aleatorios ayudan a mitigar el sobreajuste y suelen ser robustos y precisos en la clasificación.  
   Referencia: Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32

Para la selección del método a utilizar, es importante destacar que el esp32 es un microcontrolador de bajo consumo de energía y recursos limitados en comparación con sistemas más potentes como una Raspberry o una computadora convencional. Esto implica que las técnicas de análisis de datos que se pueden implementar en el esp32 deben ser adaptadas a sus capacidades y restricciones.

Por lo tanto métodos como el usado en [Tesis]; LDA con PCA, en el caso del esp32, su implementación puede ser desafiante debido a las limitaciones de memoria y capacidad de cálculo del microcontrolador. El PCA, en particular, puede requerir una gran cantidad de memoria y poder de procesamiento para calcular las componentes principales.

Por otro lado, las redes neuronales son una opción para implementar en el esp32, ya que se pueden adaptar a sus restricciones y aprovechar al máximo sus capacidades. Las redes neuronales, especialmente las redes neuronales convoluciones (CNN) y las redes neuronales recurrentes (RNN), han demostrado ser efectivas en tareas de clasificación y reconocimiento de patrones en diversos dominios.

*Referncia:*

*Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.*

Las redes neuronales también ofrecen oportunidades de optimización específicas para dispositivos de recursos limitados. Técnicas como la cuantificación de pesos y la compresión del modelo permiten reducir aún más el tamaño del modelo y los requisitos de memoria, mientras que las técnicas de aceleración de hardware como la inferencia en hardware especializado o la optimización de código pueden mejorar el rendimiento y la eficiencia energética en el ESP32.

Referencias:

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or-1. arXiv preprint arXiv:1602.02830.

Por lo tanto, el método de redes neuronales se puede implementar correctamente a una placa esp32.

**Redes Neuronales:**

Las redes neuronales son un modelo computacional inspirado en el funcionamiento del cerebro humano. Están diseñadas para aprender y reconocer patrones complejos en conjuntos de datos mediante el procesamiento de información a través de múltiples capas de neuronas interconectadas. Estas pueden resolver problemas complejos de clasificación, regresión, reconocimiento de imágenes, procesamiento de lenguaje natural

La neurona y su funcionamiento:  
La neurona es la unidad fundamental de procesamiento en una red neuronal. Se compone de tres partes principales: las dendritas, el cuerpo celular y el axón. Las dendritas reciben señales de entrada de otras neuronas o del entorno, y el cuerpo celular procesa estas señales. Luego, el axón transmite la señal procesada a otras neuronas.

El perceptrón y sus características:  
El perceptrón es un tipo básico de neurona artificial utilizado en redes neuronales. Tiene conexiones ponderadas que multiplican las señales de entrada y las suman. Luego, se aplica una función de activación para determinar si la neurona se activa o no. El perceptrón puede aprender a ajustar los pesos de las conexiones para lograr una salida deseada mediante un proceso llamado aprendizaje supervisado.

Referncias

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press

Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford University Press.

**Implemtacion de Redes Neuronales**

Para la implementación de redes neuronales, se utilizara la librería de Tensorflow y Keras, las cuales traen toda la implementación para crear redes neuronales, para entrenarlas y utilizarlas para clasificar.

**Arquitectura Red Neuronal**

Para la arquitectura interna no existen métodos específicos para se selección , pero Keras trae un nuevo método de obtención de hiperparámetros óptimos con su librería keras-tuner para Python .

Se hicieron pruebas con distintas redes neuronales como ….

Resultados…..

**Servidor**

Para poder utilizar el modelo por en instrumento de campo será necesario subir a un servidor la información del modelo de keras, , para ello se utilizara la propia plataforma de git para actualizar los modelos en la web , los modelos seran genrados por la propia librería de tensorflow , pero en su versión lite para dispositivos de menores prestaciones , para que puedan ser utilizadas estas redes entrenadas; por otros lenguajes y plataformas.

**Dispositivo de Campo**

El dispositivo de campo se encargara de tomar los modelos hechos por el dispositivo de campo , y lograr encontrar el estado de la leche , tiene que ser un dispositivo que e pueda comunicar con el sensor, que pueda contarse al servidor online, descargar los modelos y almacenarlos , además de ser un dispositivo configurable , para ello se utilizara el esp32.

**Lenguaje**

Esp32 sera programado en C++ sobre la placa de Arduino , programado con el ide de VSC en PlattaformIO.

**Interfaz Visual**

El esp32 implantará una interfaz visual para permitir actualizar los modelos de