

Universidade Federal de Itajubá - UNIFEI

Monitoramento Acústico de Fluxo Hídrico

TinyML - Aprendizado de Máquina Aplicado Para Dispositivos IOT Embarcados



Bruno Vilela Batista - 2020003845

Felipe Souza Fernandes - 2020006453

# 1. Introdução

A medição da chuva é uma prática essencial no campo da meteorologia e possui um papel crucial na compreensão e gestão dos recursos hídricos. Esta atividade é vital para a previsão do tempo, estudos climáticos e a gestão de bacias hidrográficas. O monitoramento preciso da precipitação pluvial fornece dados indispensáveis para diversas áreas, incluindo agricultura, planejamento urbano, controle de enchentes e secas, além da formulação de políticas ambientais e de desenvolvimento sustentável.

A importância da medição da chuva se estende à capacidade de prever e mitigar desastres naturais. Enchentes e deslizamentos de terra, que frequentemente resultam em perdas humanas e materiais significativas, podem ser melhor previstos e gerenciados com base em dados pluviométricos confiáveis. Da mesma forma, a compreensão das variações sazonais e anuais da precipitação é fundamental para o planejamento agrícola, permitindo uma gestão mais eficiente dos recursos de irrigação e contribuindo para a segurança alimentar.

## 2. Objetivos

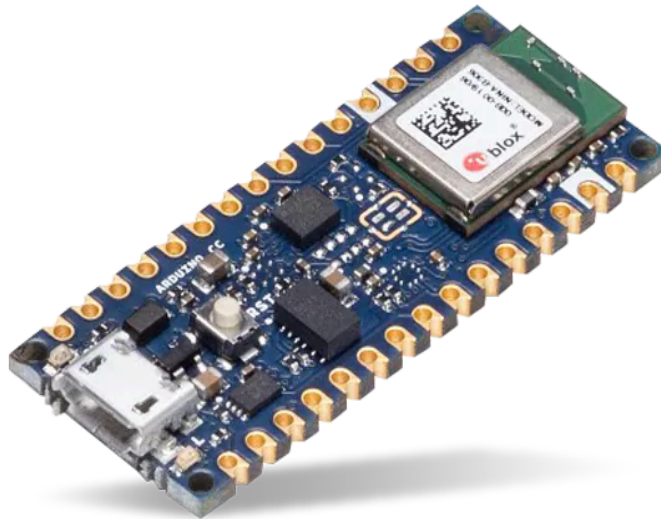
O presente relatório tem como objetivo principal analisar e aprimorar os métodos de medição da chuva, destacando sua importância para a previsão do tempo, gestão de recursos hídricos e mitigação de desastres naturais. Para isso, um dos objetivos centrais deste estudo é desenvolver um modelo de machine learning capaz de identificar a intensidade da chuva através do som, utilizando um chuveiro para simular a precipitação. Este modelo representa um passo inicial para um estudo futuro que visa a utilização de dispositivos TinyML em estações pluviométricas, com o propósito de reduzir a quantidade de componentes móveis. Assim, buscamos explorar novas maneiras eficientes de medição de chuva, contribuindo para uma gestão mais eficaz dos recursos ambientais e facilitando a adaptação às mudanças climáticas.

## 3. Descrição do Projeto

Hardware e Software voltados para a medição de chuvas, utilizando inicialmente o chuveiro em fluxos alternados para simular diferentes níveis de precipitação.

### 3.1 Hardware

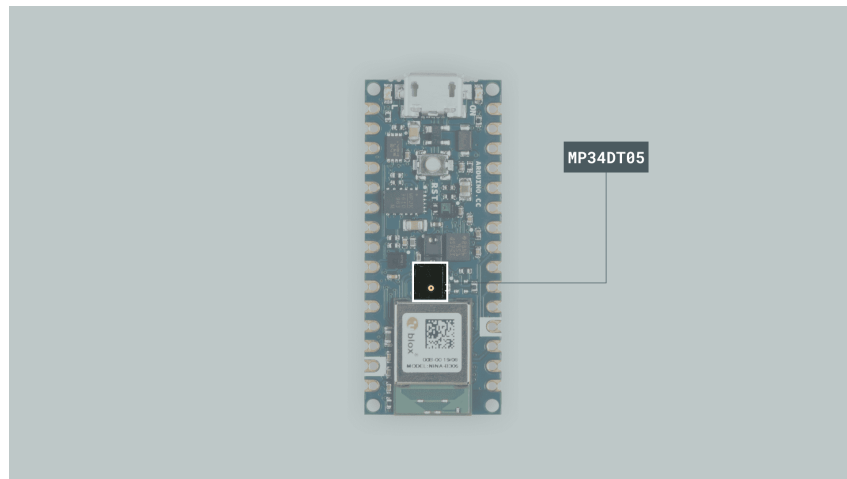
O principal componente utilizado no projeto foi o **Arduino Nano 33 BLE Sense**, uma placa de desenvolvimento que oferece uma combinação poderosa de recursos em um formato compacto. Esta placa é ideal para projetos que exigem processamento de sinais de sensores em tempo real e implementação de inteligência artificial embarcada. Ela se destaca por sua capacidade de executar modelos de IA utilizando TinyML e TensorFlow™ Lite, tornando-a uma ferramenta versátil para aplicações avançadas em IoT e sistemas embarcados.



*Imagem 1: Arduino Nano 33 BLE Sense*

Um dos sensores que compõem essa placa é o sensor **MP34DT05**, um microfone ultra compacto que utiliza PDM (Pulse-Density Modulation) para representar um sinal analógico com um sinal binário. A faixa de valores diferentes do sensor é a seguinte:

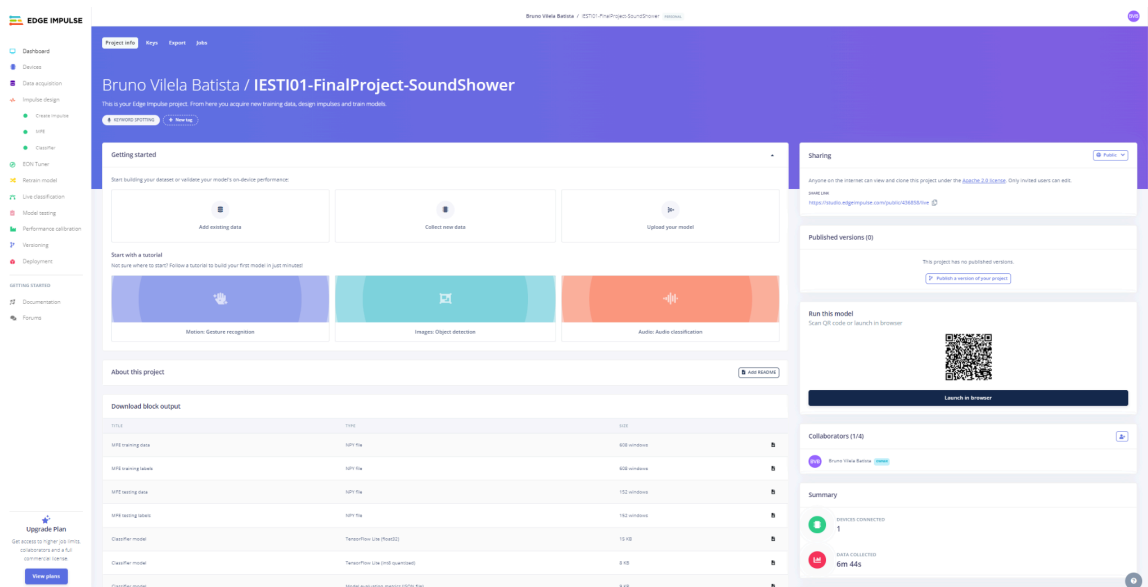
- Relação sinal-ruído: 64dB
- Sensibilidade: -26dBFS  $\pm$ 3dB
- Faixa de temperatura: -40 a 85°C



*Imagem 2: MP34DT05 microphone sensor*

### 3.2 Software

A plataforma digital de desenvolvimento utilizada foi a Edge Impulse, que automatiza diversas tarefas essenciais no processo de machine learning. Com a Edge Impulse, é possível capturar, processar e analisar dados de sensores de maneira eficiente, além de treinar e implementar modelos de aprendizado de máquina diretamente no Arduino Nano 33 BLE Sense. Esta plataforma simplifica a criação de soluções inteligentes, permitindo o desenvolvimento rápido e preciso de aplicações embarcadas com inteligência artificial.



*Imagem 3: Dashboard do projeto no Edge Impulse*

Na etapa de deploy uma das opções possíveis é a geração de uma biblioteca Arduino, uma opção agradável pois permite a personalização do código gerado e a carga dele em

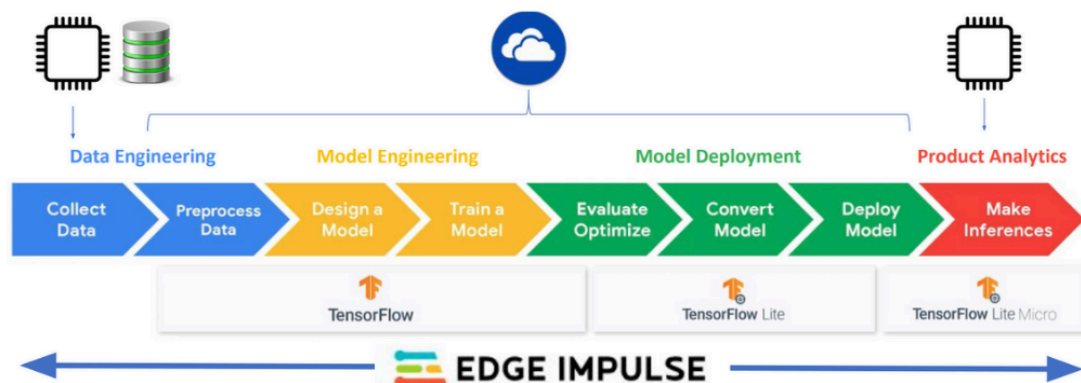
diversos dispositivos compatíveis. Para personalizar e carregar o código a plataforma arduino IDE também é uma boa solução, proporcionando formas práticas para realizar tais ações.



*Imagem 4: Arduino IDE*

### 3.3 Diagrama de blocos

O fluxo de trabalho seguido neste projeto é ilustrado pelo diagrama de blocos abaixo, representando o padrão para o desenvolvimento de projetos de machine learning. Vale destacar que a Edge Impulse fornece uma solução completa que abrange todas as etapas desse processo.



*Imagem 5: Ml workflow*

## 4. Coleta de Dados

Através da interface disponibilizada pelo Edge Impulse, na aba Data Acquisition, foi feita uma coleta de dados utilizando o microfone do Arduino. Foram escolhidas quatro

classes para simular a intensidade de precipitação através do chuveiro, são elas: Noise, Weak, Medium, Strong.

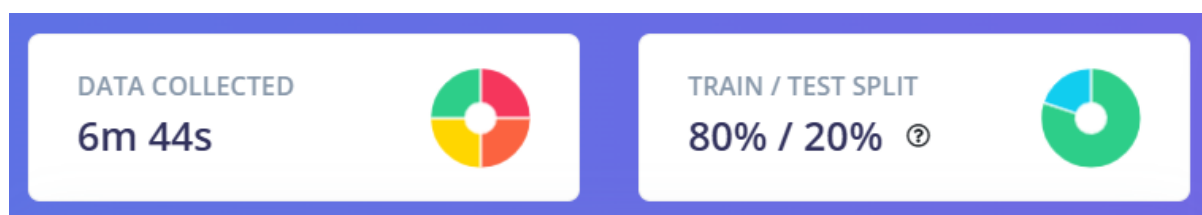
- Noise - Chuveiro desligado e som ambiente.
- Weak- Chuveiro com vazão baixa, apenas algumas gotas.
- Medium - Chuveiro com vazão média.
- Strong - Chuveiro com vazão alta.

A fim de proteger os componentes de qualquer contato com a água, foi criada uma estrutura rudimentar composta pelo Arduino, um copo, no qual o componente elétrico foi fixado através de fita isolante, e ambos foram inseridos em um recipiente arredondado, que serviria como proteção externa. A estrutura pronta foi então colocada no box do banheiro, embaixo do chuveiro, e variou-se a intensidade do mesmo.



*Imagem 6: Encapsulamento*

Para cada classe foram coletados dez áudios de dez segundos cada, sendo oito para treinamento e dois para teste, criando, então, uma proporção de 80/20. A frequência utilizada para capturar os dados foi de 16 KHz, respeitando as configurações do Arduino.



*Imagem 7: Dados coletados*

## 5. Modelos

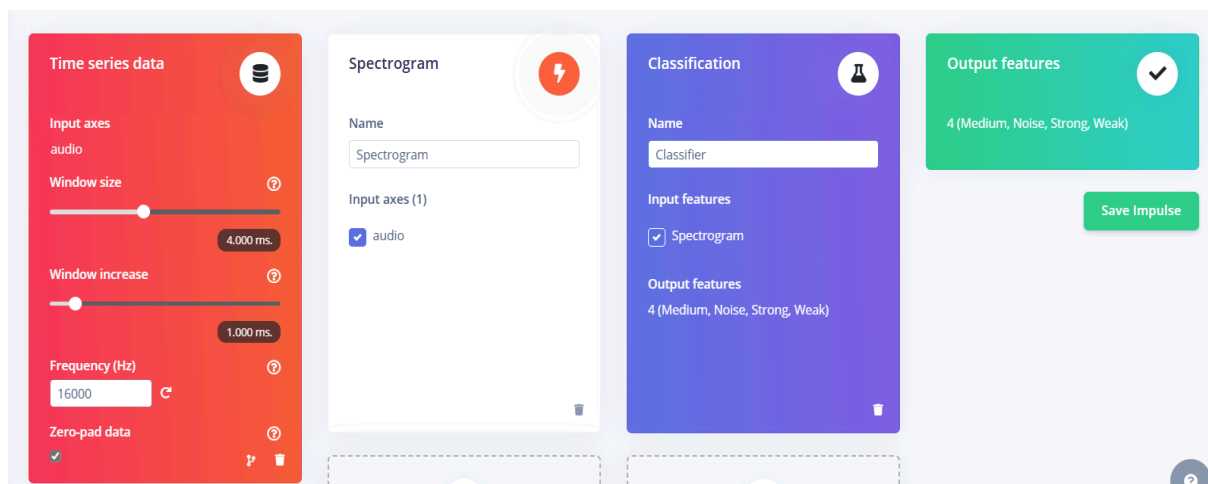
Com os dados coletados, iniciou-se a busca por um modelo que melhor interpretasse os dados e fosse capaz de fazer inferências com uma boa acurácia. Para isso, foram testados dois modelos, o primeiro utilizando espectrogramas e o segundo MFE, ambos blocos de processamento disponibilizados pelo Edge Impulse.

O espectrograma é um bloco de processamento que extrai características de tempo e frequência de um sinal. Ele funciona bem com dados de áudio para casos de uso não relacionados ao reconhecimento de voz, ou com quaisquer dados de sensores que possuam frequências contínuas. O bloco, primeiramente, divide a janela em vários quadros sobrepostos, em seguida, uma FFT é calculada para cada quadro e, finalmente, o valor do Piso de Ruído (Noise floor) é aplicado ao espectro de potência. As características geradas pelo bloco de espectrograma são iguais ao número de quadros temporais gerados vezes o número de características de frequência.

Já o bloco de processamento MFE, semelhante ao de espectrograma, extrai características de tempo e frequência de um sinal. No entanto, ele usa uma escala não linear no domínio da frequência, chamada escala Mel. Ele funciona bem com dados de áudio, principalmente para casos de uso não relacionados ao reconhecimento de voz, quando os sons a serem classificados podem ser distinguidos pelo ouvido humano. A extração de características é semelhante à do bloco de espectrograma, mas adiciona 2 etapas extras. Após calcular o espectrograma, filtros triangulares são aplicados em uma escala Mel para extrair bandas de frequência. A escala Mel é uma escala perceptual de tons julgados pelos ouvintes como equidistantes uns dos outros. A ideia é extrair mais características (mais bancos de filtros) nas frequências mais baixas e menos nas frequências mais altas, assim, ela funciona bem em sons que podem ser distinguidos pelo ouvido humano. A última etapa limpa a saída do MFE para redução de ruído. Qualquer amostra abaixo do Piso de Ruído é definida como zero.

### 5.1 Modelo Espectrograma

O primeiro modelo gerado utilizou espectrogramas e pode ser visto a seguir.



*Imagem 8: Bloco espectrograma*

Utilizou-se um tamanho de janela de quatro segundos, um aumento de janela de um segundo e frequência de 16 KHz. Como foi dito, optou-se por trabalhar com espectrogramas. Por fim, foi adicionado o bloco de classificação.

Na aba “Spectrogram”, os parâmetros escolhidos foram os seguintes.

Parameters		Autotune parameters
Spectrogram		
Frame length ?	0.05	
Frame stride ?	0.05	
FFT length ?	128	
Normalization		
Noise floor (dB) ?	-32	
		Save parameters

*Imagem 9: Parâmetros modelo 1*



Com esses parâmetros decididos, obteve-se um gráfico com a distribuição de cada classe. É possível observar uma boa separação entre as classes.

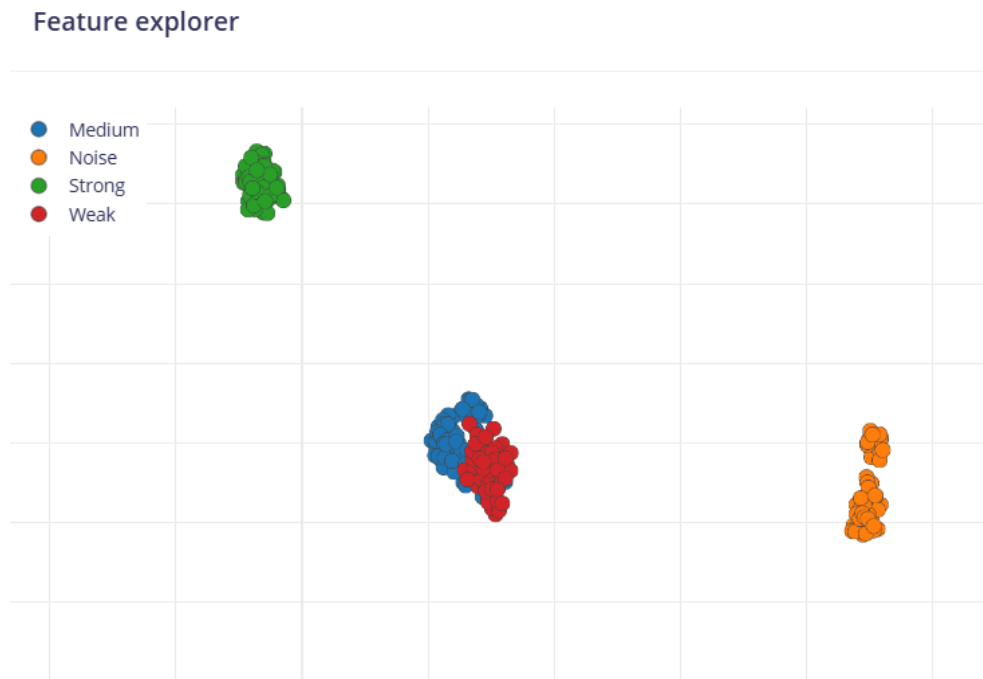


Imagem 10: Distribuição de classes modelo 1

Por fim, na aba “Classifier” foi criado o modelo em si, como pode ser observado a seguir.

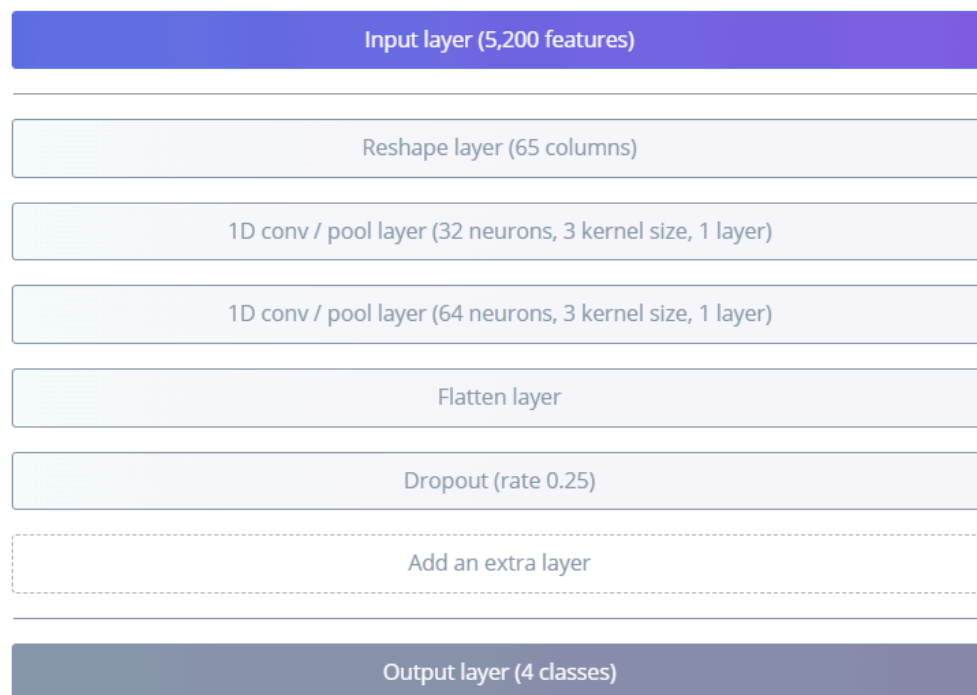
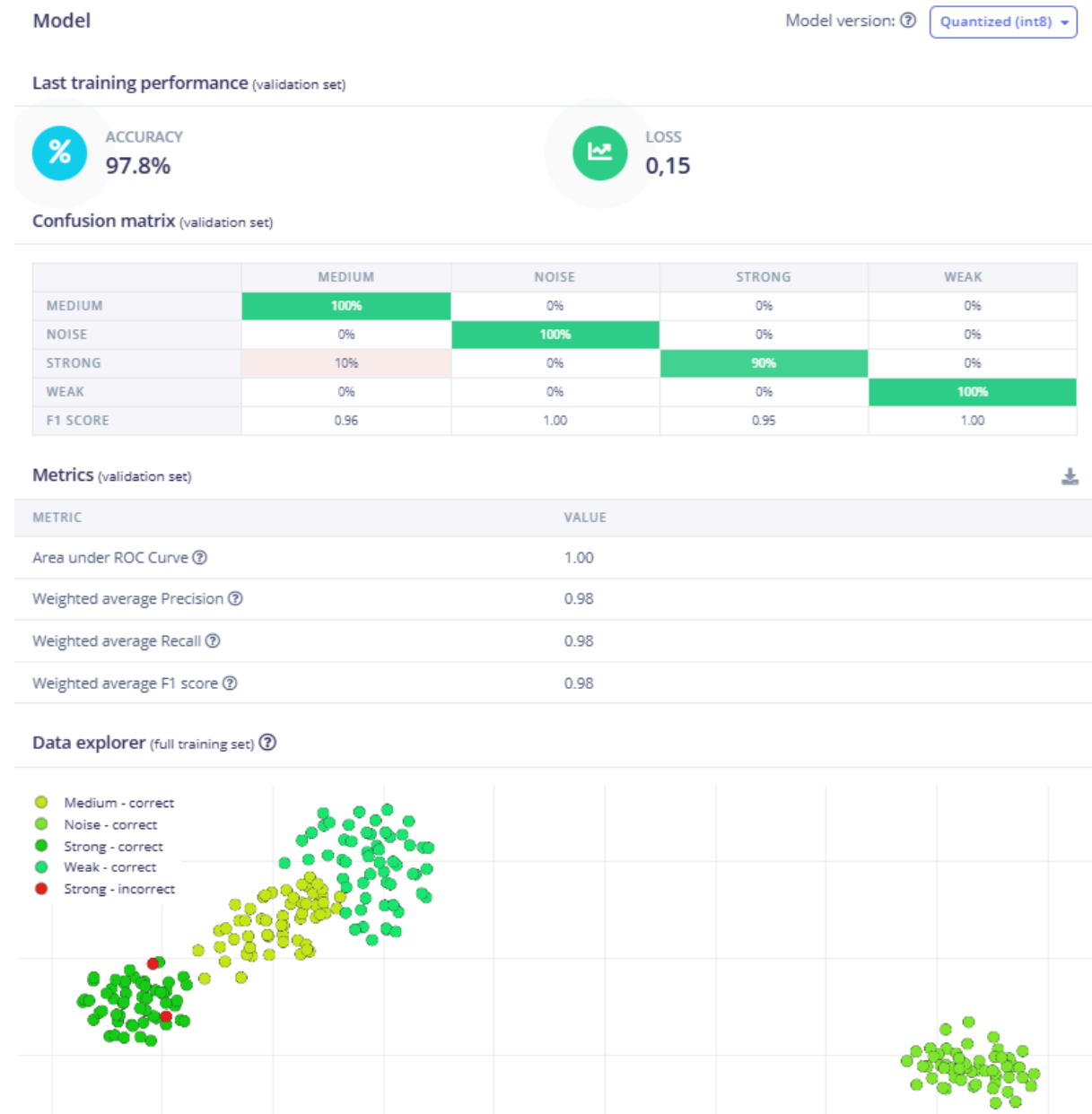


Imagem 11: Camadas modelo 1

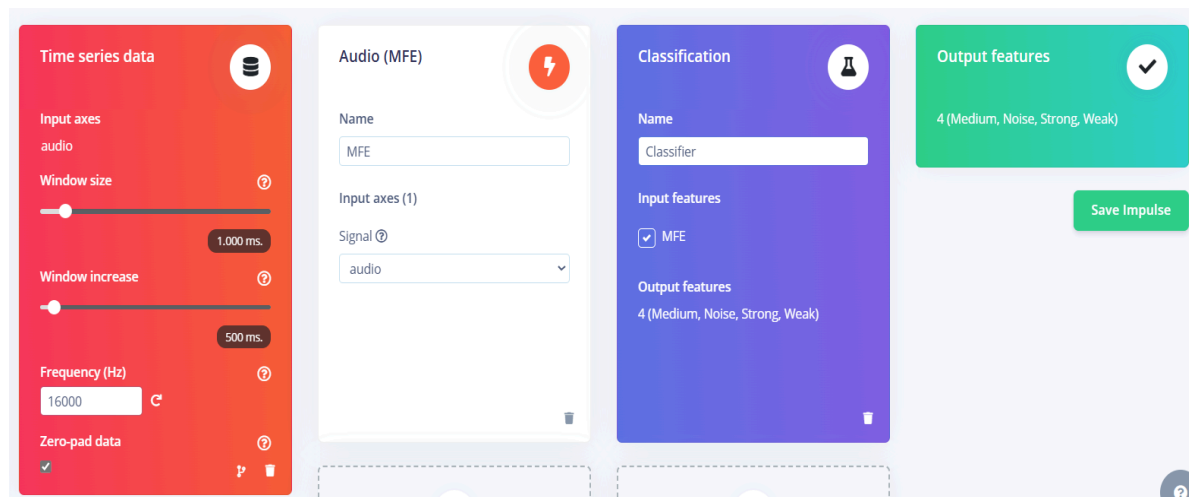
Essa configuração foi obtida através de testes e foi a que obteve a maior acurácia na fase de treinamento. Para o treinamento, utilizou-se um número de ciclos de 100 e uma taxa de aprendizado de 0,005. Com esse modelo e esses parâmetros a acurácia atingiu um valor de 97,8% e uma perda de 0,15.



*Imagem 12: Resultado treinamento modelo 1*

## 5.2 Modelo MFE

Já no segundo modelo foi utilizado o bloco MFE.



*Figura 13: Bloco MFE*

Utilizou-se um tamanho de janela de um segundo, um aumento de janela de meio segundo e frequência de 16 KHz. Adicionou-se o bloco MFE e por fim, o bloco de classificação.

Na aba “MFE” escolheu-se os seguintes parâmetros.

Parameters
Autotune parameters

Mel-filterbank energy features

Frame length ?	0.02
Frame stride ?	0.01
Filter number ?	40
FFT length ?	256
Low frequency ?	0
High frequency ?	Click to set

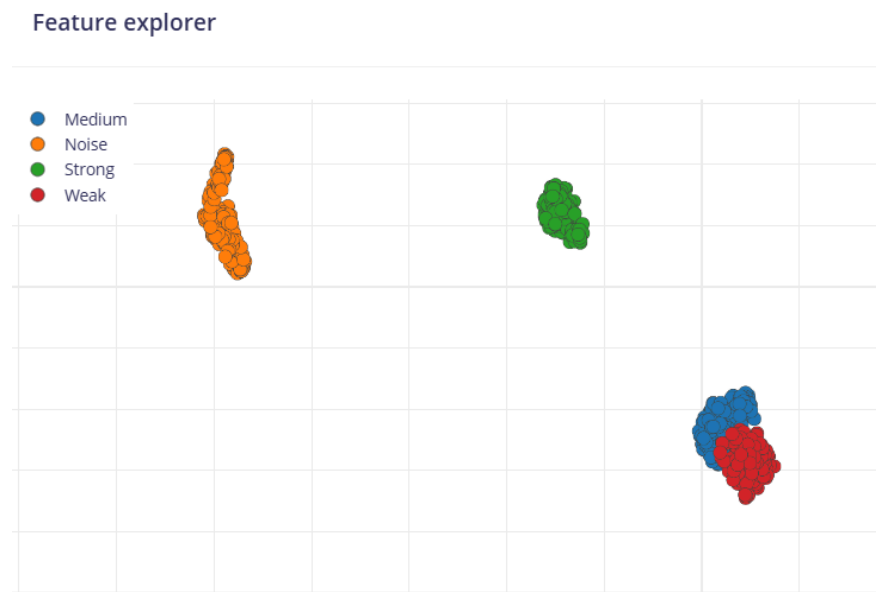
Normalization

Noise floor (dB) ?	-52
--------------------	-----

Save parameters

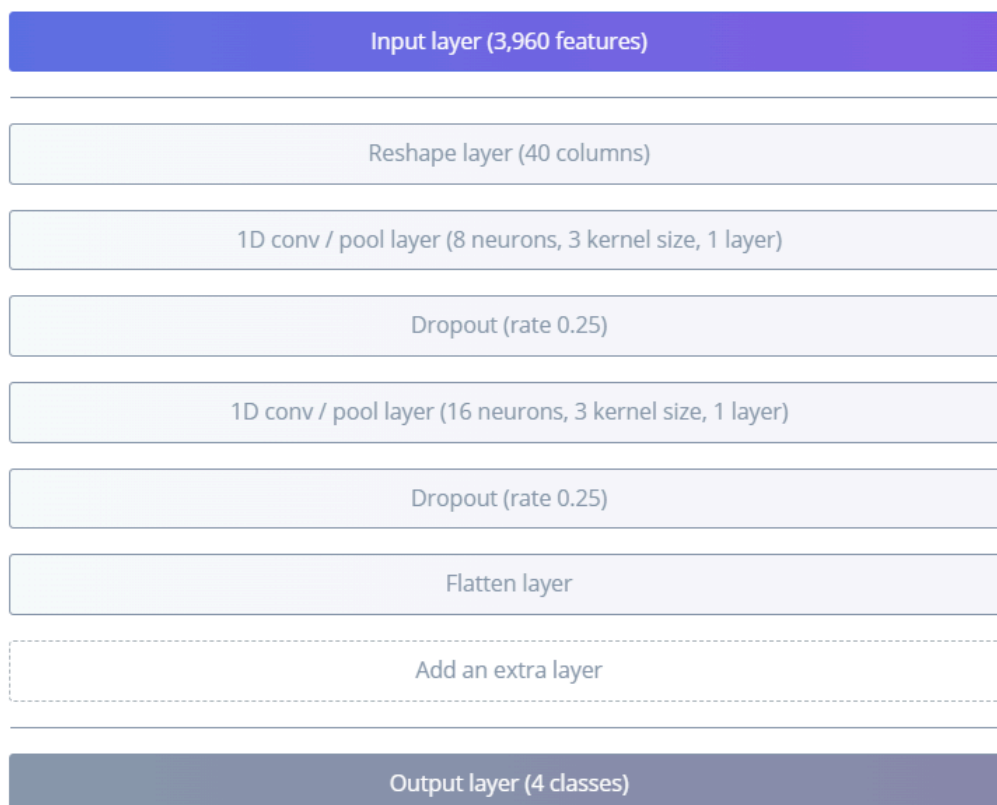
*Imagem 14: Parâmetros modelo 2*

O gráfico obtido através desses parâmetros pode ser visto abaixo. Percebe-se que, semelhante ao modelo anterior, também é possível diferenciar bem as classes.



*Imagem 15: Distribuição de classes modelo 2*

Finalmente, na aba “Classifier” é criado o modelo e é realizado o treinamento.



*Imagem 16: Camadas modelo 2*

Semelhante ao anterior, essa foi a configuração que obteve a maior acurácia. O número de ciclos foi novamente 100 e a taxa de aprendizagem 0,005. Iniciou-se então o treinamento e foi obtida uma acurácia de 100% e uma perda de 0,01.

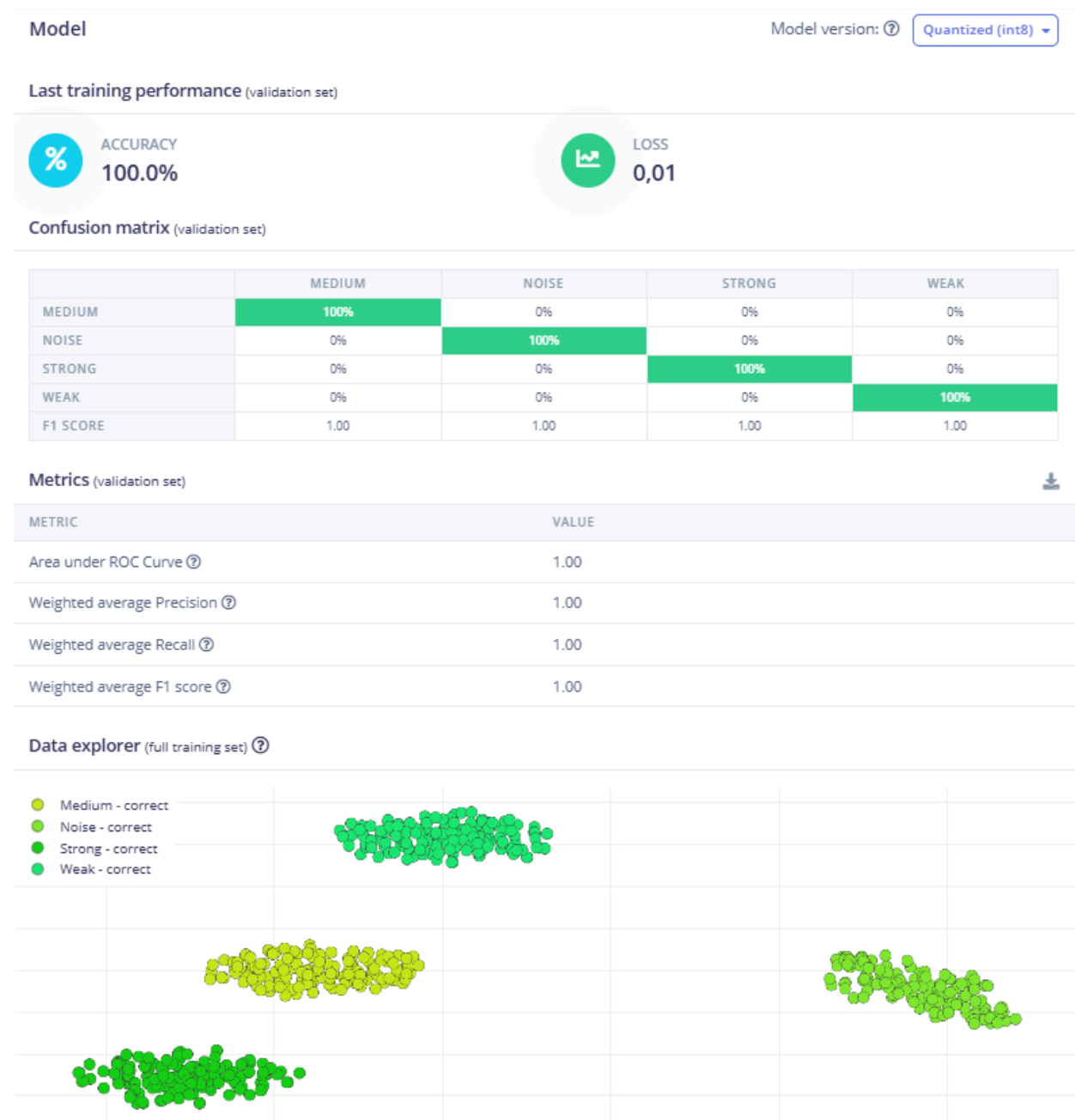
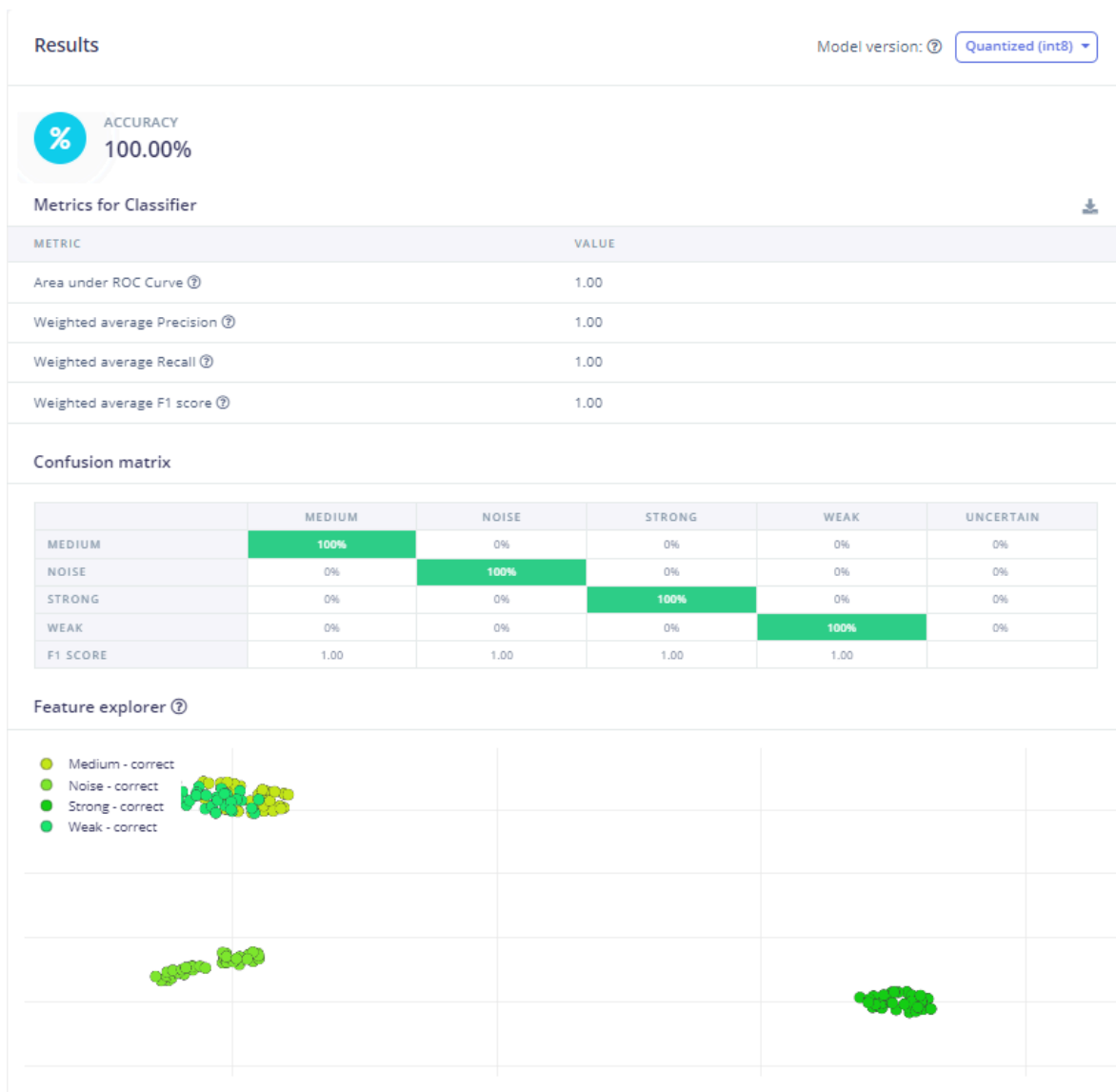


Imagem 17: Resultado treinamento modelo 2

## 6. Testes e Deploy

Primeiramente, foram utilizados os 20% de dados reservados para teste para avaliar os modelos gerados. O modelo com pré-processamento por MFE obteve resultados excelentes, alcançando 100% de acurácia em todas as classes.



*Imagem 18: Resultados dos testes - modelo MFE*

Os resultados dos testes com o pré-processamento utilizando espectrograma também foram muito bons, apresentando uma acurácia média de 92,86%. No entanto, esse desempenho é significativamente inferior ao do modelo baseado em MFE.

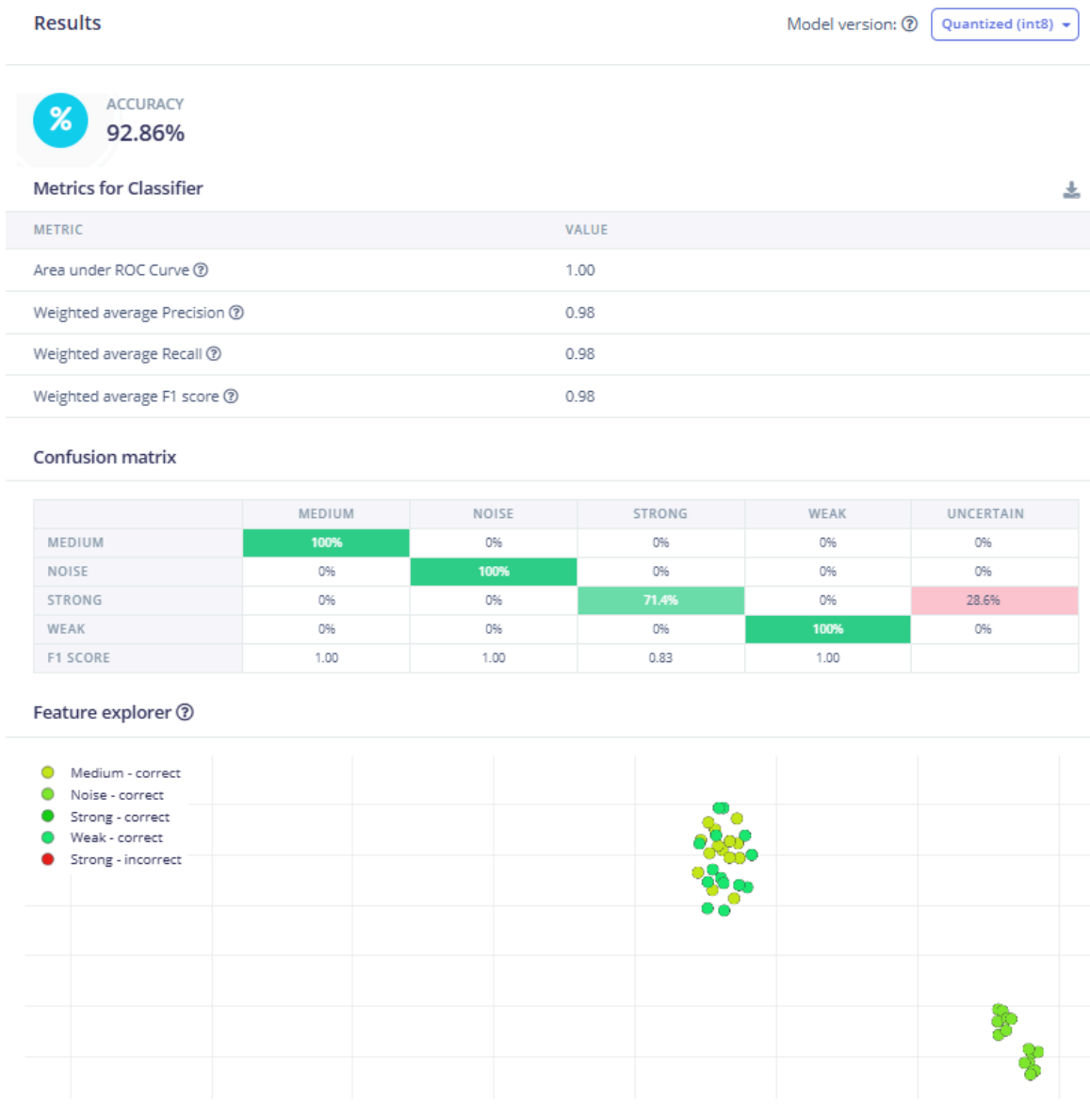


Imagem 19: Resultados dos testes - modelo Espectrograma

Após realizar os testes na plataforma Edge Impulse, o modelo utilizando o pré-processamento por MFE foi selecionado para ser carregado na placa Arduino. O método de deploy escolhido foi a geração de uma biblioteca Arduino, uma opção recomendada por permitir facilidade de personalização e versatilidade na escolha de dispositivos compatíveis. Concluídas as alterações necessárias no código para controlar o LED RGB em função da classe detectada, o modelo foi introduzido na memória da placa, capacitando-a a realizar inferências sobre o áudio captado pelo microfone. Por fim, a placa e o encapsulamento foram colocados novamente sob a queda da água proveniente do chuveiro e as inferências foram realizadas.



*Imagem 20: inferências - desligado(noise), verde(weak), azul(medium), vermelho(strong)*

## 7. Conclusão

A criação de um modelo de machine learning utilizando TinyML para simular e classificar a intensidade da chuva através de um chuveiro demonstrou resultados excelentes. Este estudo comprovou a eficácia de modelos compactos e eficientes, capazes de operar em dispositivos com recursos limitados, como microcontroladores. A implementação bem-sucedida destaca o potencial do TinyML em aplicações práticas e acessíveis, abrindo novas possibilidades para a detecção e classificação de padrões em diversos campos.

A precisão alcançada na simulação da intensidade da chuva reforça a viabilidade dessa abordagem, oferecendo uma solução promissora e econômica para a monitoração ambiental e outras aplicações sensoriais. Além disso, torna possível a utilização de TinyML para a medição pluviométrica em estações meteorológicas, proporcionando um método eficiente e de baixo custo para a coleta de dados essenciais na análise climática e na gestão de recursos hídricos.

<https://studio.edgeimpulse.com/public/437196/live> (Projeto - Espectrograma)

<https://studio.edgeimpulse.com/public/436858/live> (Projeto - MFE)