

Análise em tempo e frequência para extração de características de sinais obtidos de acelerômetros triaxiais com aplicação no estudo da marcha

Aluno: Rafael Shibana Fayan

Orientador: Moacir Antonelli Ponti

Motivação e Contextualização

Preocupação com Segurança e Saúde da Terceira Idade

- Com o envelhecimento da população mundial, uma maior atenção está sendo dada aos cuidados voltados para idosos
- Uma das maiores preocupações é o risco de queda devido a alta taxa de incidência e severidade das consequências
- É de grande interesse da comunidade investigar padrões para rastreamento do risco de queda através de testes acessíveis.

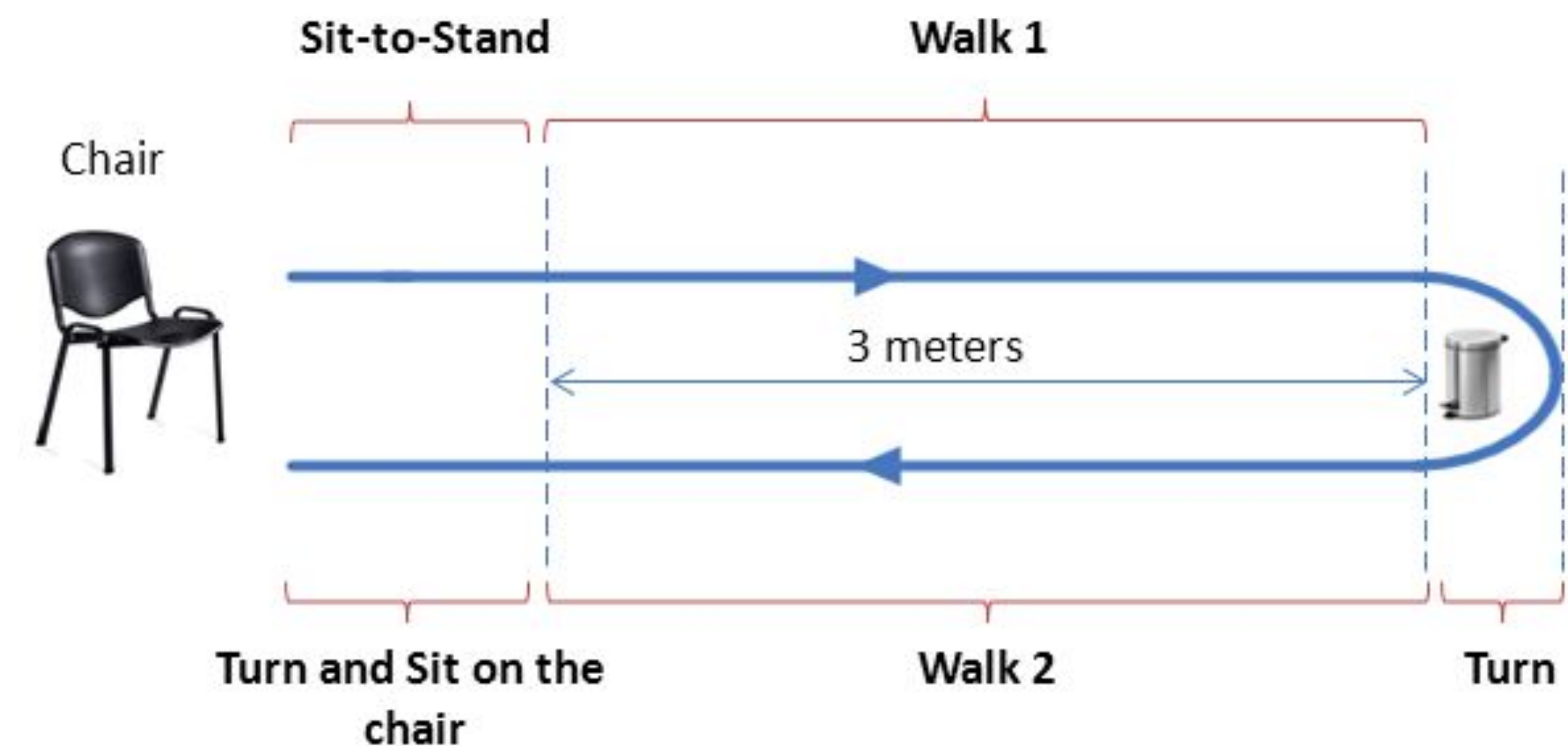
Objetivos

Realizar uma análise em frequência sob dados de aceleração coletados durante o teste Timed Up and Go de um conjunto de voluntários

- Será feita a geração dos espectrogramas dos sinais de aceleração, extração de características a partir de autoencoders e classificação dos pacientes entre caidores e não-caidores com algoritmos de aprendizado de máquina
- Ao final do projeto, espera-se obter padrões que evidenciem a diferença entre caidores e não caidores, permitindo a classificação de idosos entre perfis de baixo e alto risco de queda

Timed Up and GO (TUG)

- Teste clínico para avaliação da mobilidade e equilíbrio do paciente durante marcha e mudanças de posturas
- Para o projeto, utilizou-se como base de dados medições de aceleração de 79 voluntários acima de 60 anos obtidos durante 9 testes TUG executados sequencialmente.





Desenvolvimento e Atividades Realizadas

Preparação do ambiente

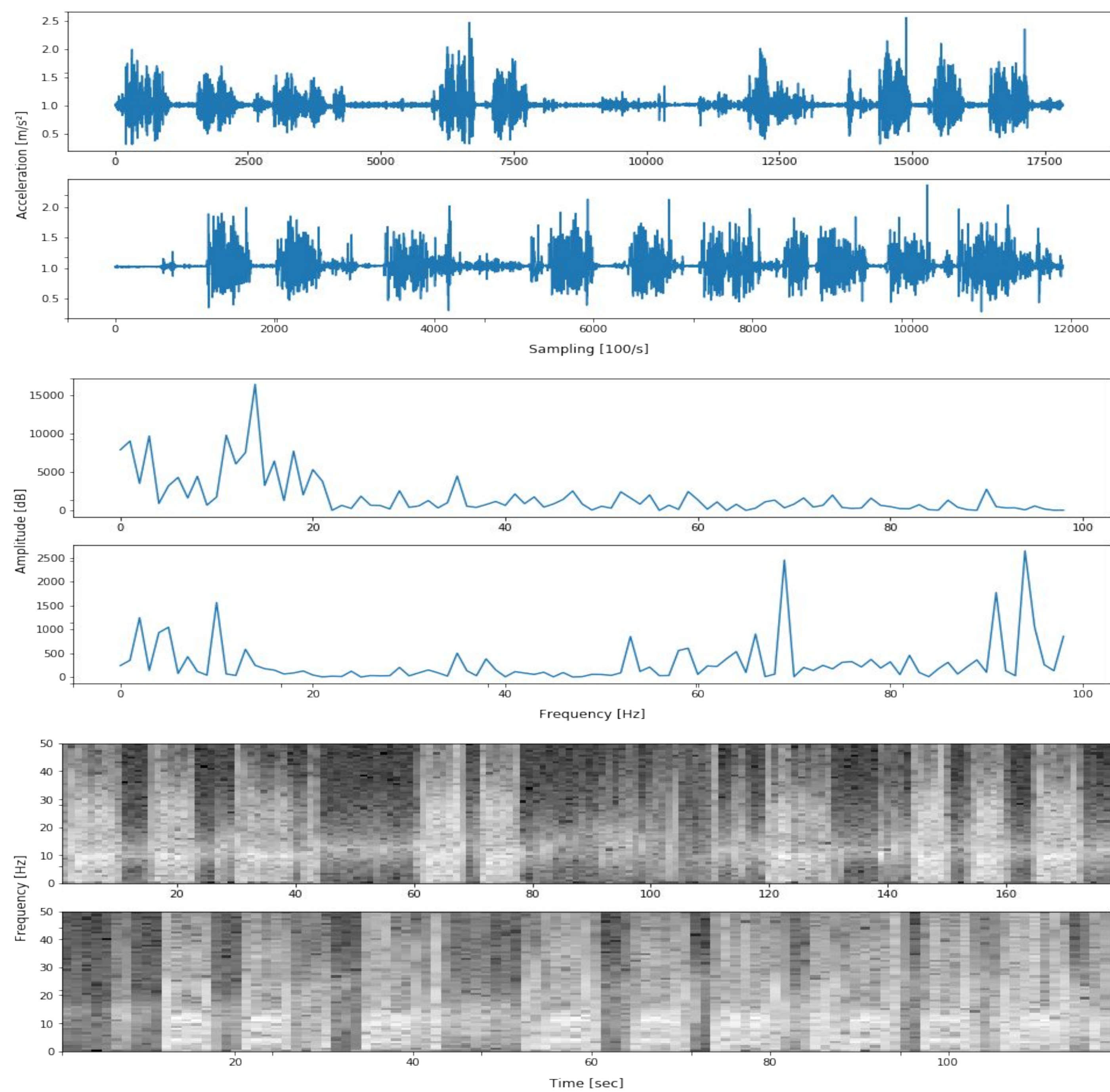
- Escolheu-se a linguagem python em conjunto com o framework Jupyter para o desenvolvimento dos algoritmos
- Criou-se um repositório github para versionamento do código e futuras colaborações
- Para fins de comparação, extraiu-se características a partir dos dados de aceleração convertidos para o espectro da frequência

Geração dos Espectrogramas

- A partir dos dados de aceleração para cada paciente, aplicou-se a Transformada de Fourier Discreta para geração dos espectrogramas

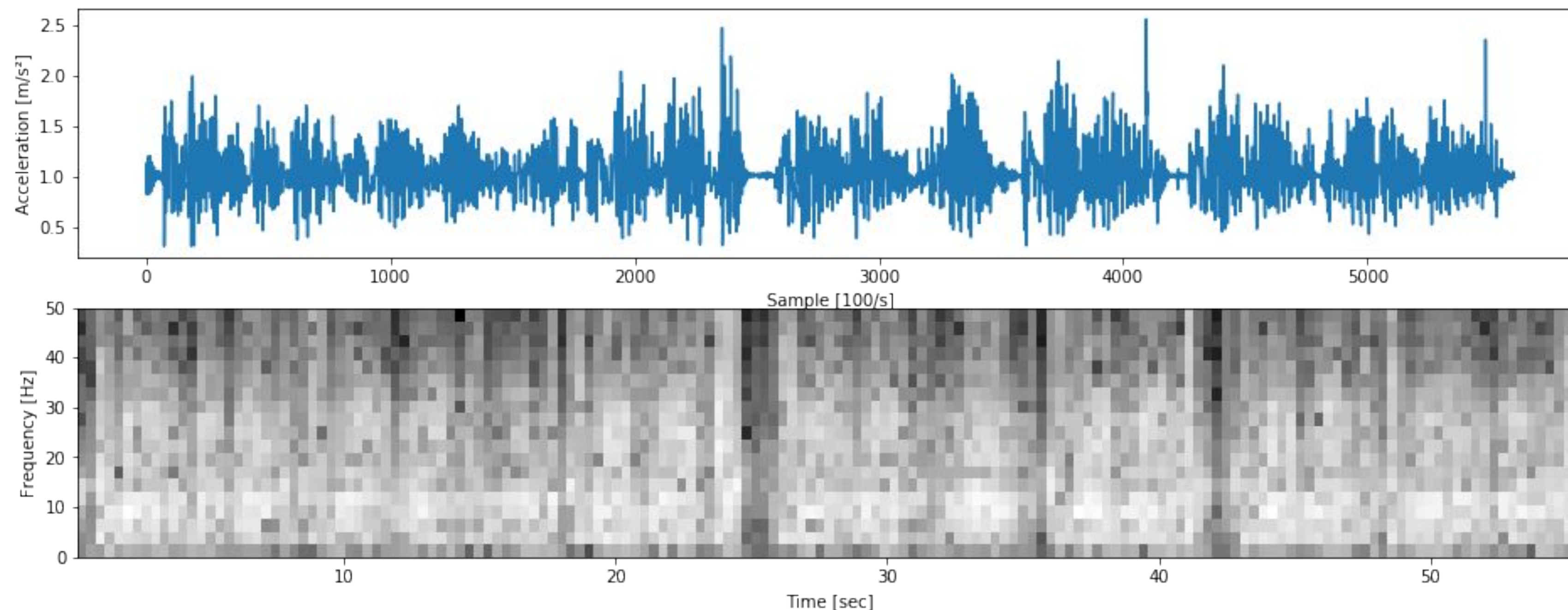
$$X_k = \sum_{n=0}^{+N-1} x[n] e^{\frac{-j2\pi kn}{N}} \quad \text{para } k = 0, 1, \dots, N-1$$

- Os espectrogramas são representações visuais do sinal que fornecem 3 informações: tempo ou amostragem, frequência e amplitude.



Geração dos Espectrogramas

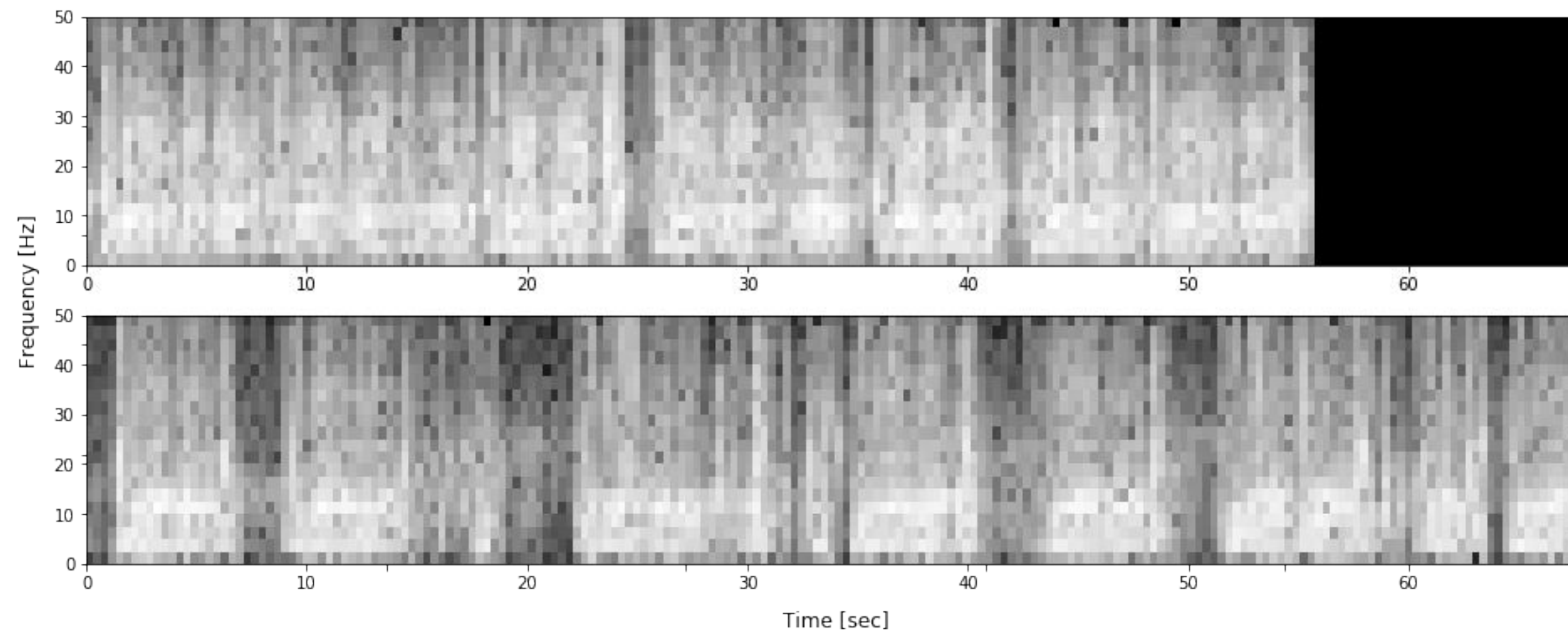
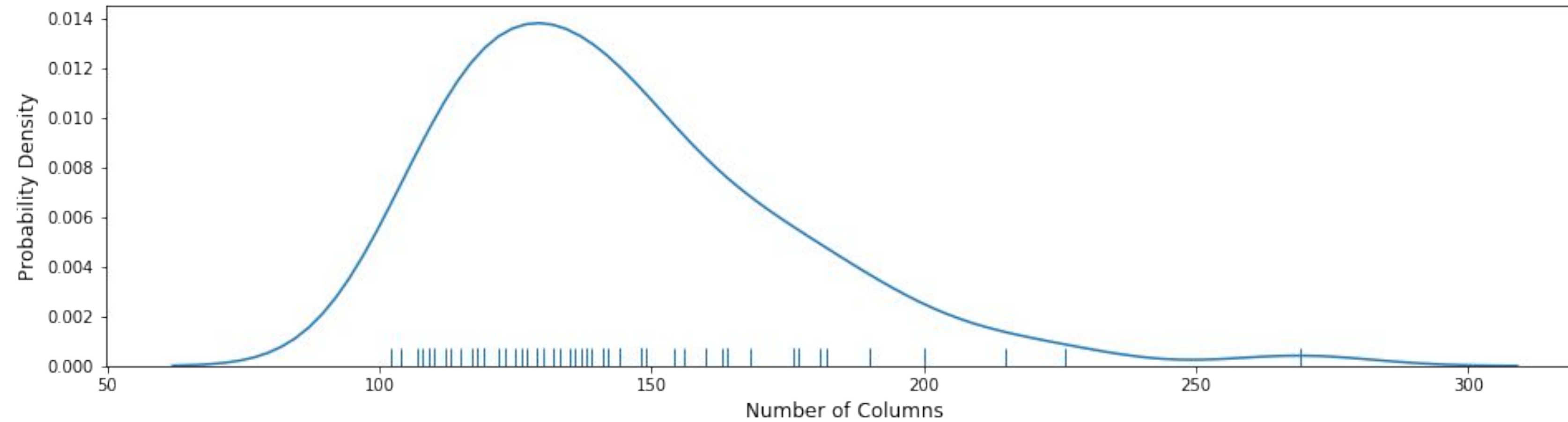
- Além dos espectrogramas gerados a partir dos dados de aceleração por completos, foram gerados os espectrogramas dos dados de aceleração correspondentes aos testes TUG concatenados



Geração dos Espectrogramas

- Para extração de *features* dos espectrogramas com autoencoders, deve-se garantir que todos possuam as mesmas dimensões
- Logo, temos de truncar ou preencher a matriz de valores dos espectrogramas
- Para os espectrogramas do sinal completo, optou-se por truncar a matriz a partir de um limiar, e para os espectrogramas do teste TUG preencheu-se as matrizes com valor 0.

Geração dos Espectrogramas



Construção dos Autoencoders

- Para extração de características dos espectrogramas, utilizou-se Autoencoders
- Um autoencoder é uma rede neural cujo objetivo é reproduzir um dado de entrada a partir de um conjunto de informações extraídos dele

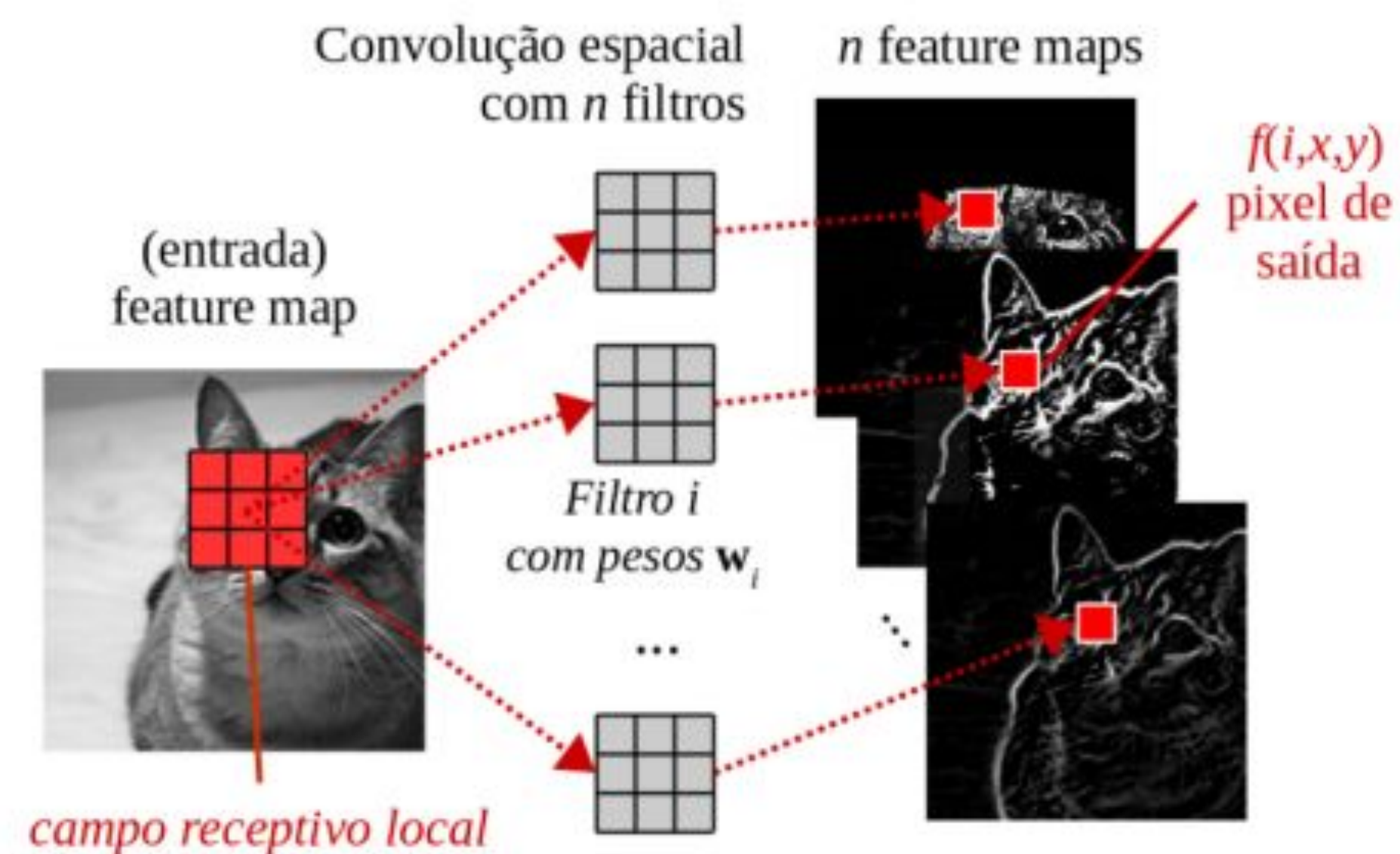
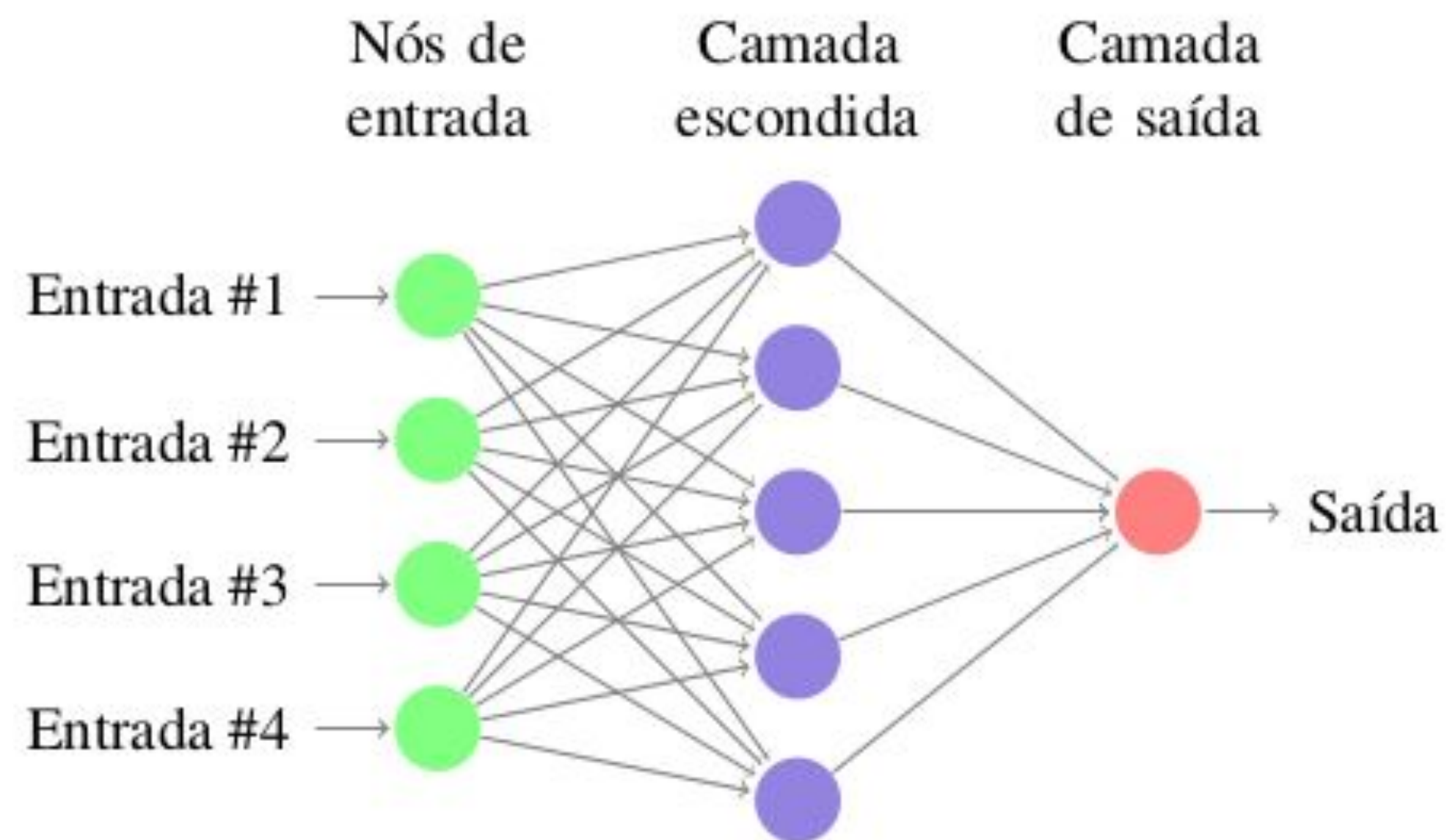


- Resumidamente, ele busca minimizar a seguinte função:

$$\ell(x) = ||x - \hat{x}||^2$$

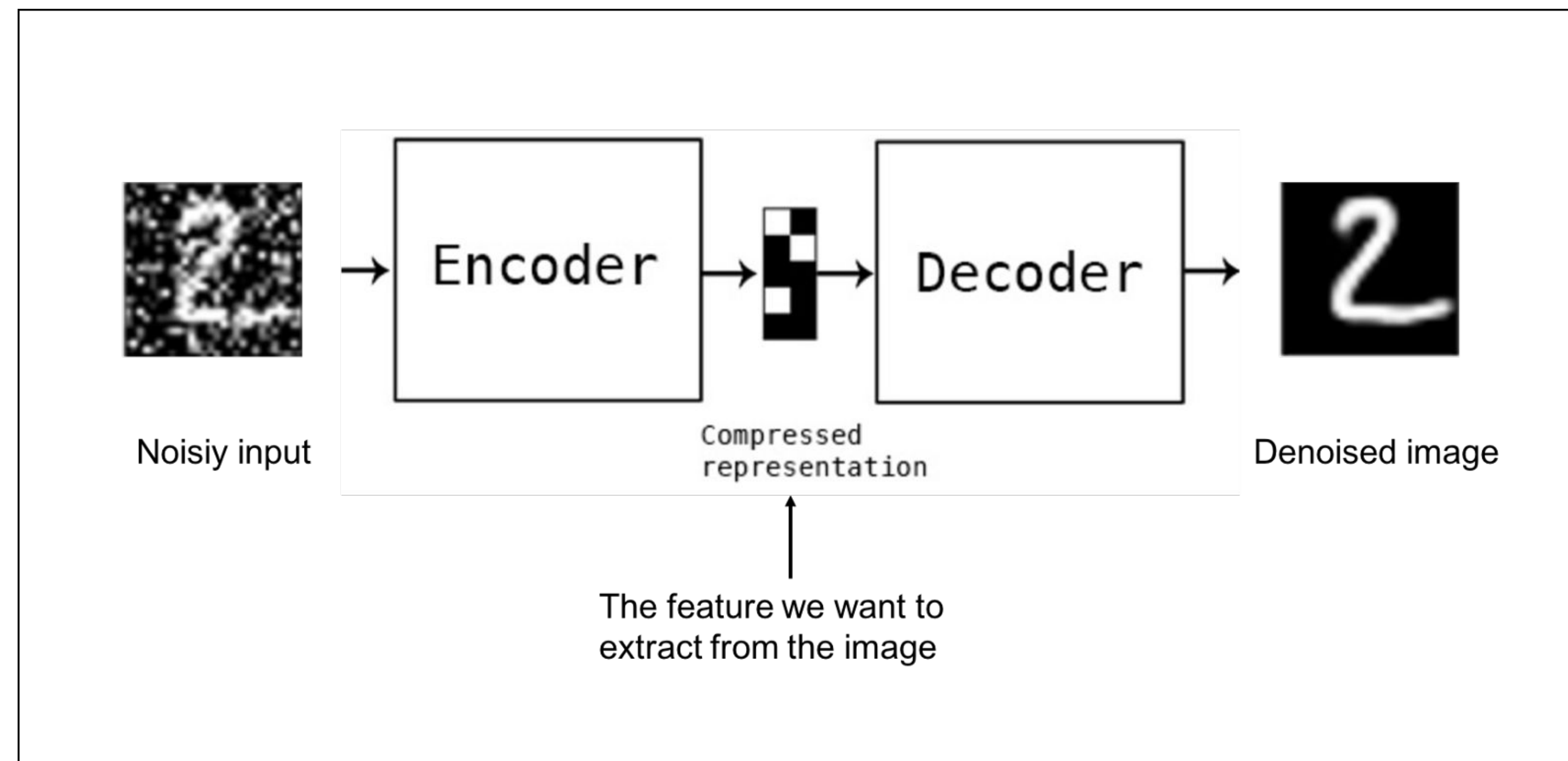
Construção dos Autoencoders

- Os autoencoders podem possuir camadas densas ou convolucionais



Construção dos Autoencoders

- A fim de obter características mais representativas e deixar o autoencoder mais robusto, aplicou-se a técnica de *denoising autoencoder*



Construção dos Autoencoders

- Utilizando a biblioteca Tensorflow com a API de alto nível Keras, foram implementadas um total de 8 arquiteturas de autoencoders:
 - 5 arquiteturas para os sinais completos
 - 1 arquitetura totalmente densa (código de 80 neurônios)
 - 1 arquitetura mista (código de 128 neurônios)
 - 3 arquiteturas totalmente convolucionais (código de 272, 544 e 1088 neurônios)
 - 3 arquiteturas para os sinais dos testes TUG concatenados
 - 1 arquitetura mista (código de 128 neurônios)
 - 2 arquiteturas totalmente convolucionais (código de 250 e 500 neurônios)

Construção dos Autoencoders

- Foram obtidos as seguintes métricas de treino para cada um dos autoencoders:

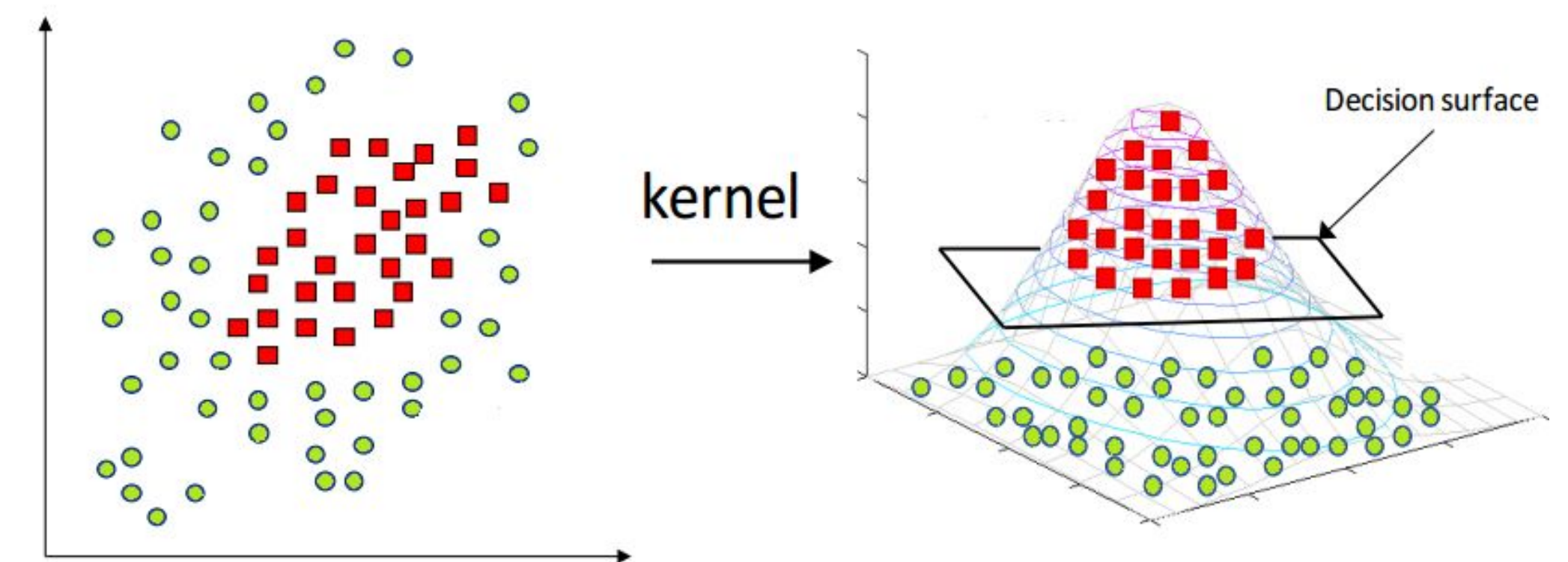
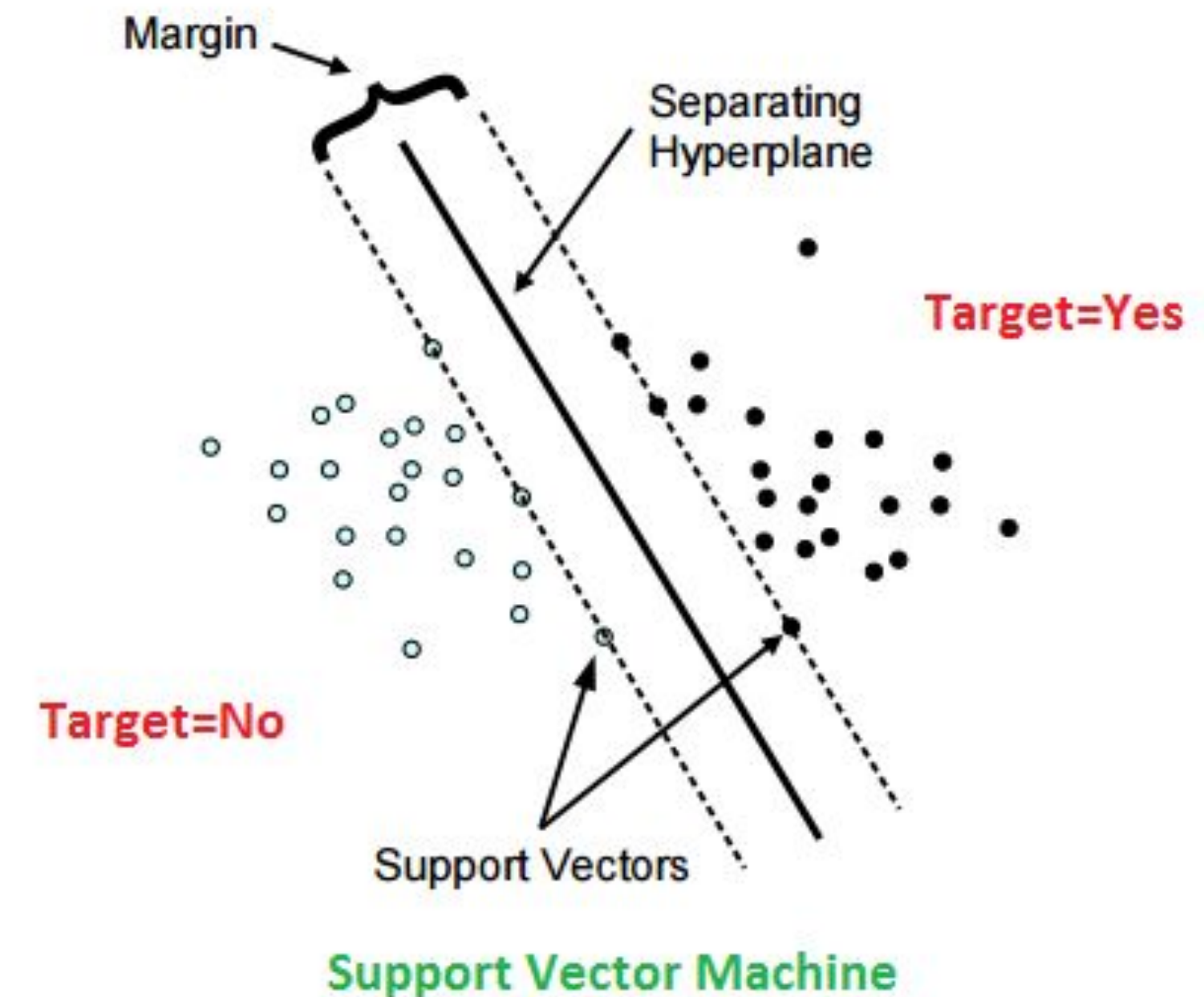
(Sinal - Características)	Loss	Validation Loss
Todo Sinal - 128 Características	0.0122	0.0170
Todo Sinal - 272 Características	0.0091	0.0108
Todo Sinal - 544 Características	0.0068	0.0073
Todo Sinal - 1088 Características	0.0060	0.0062
TUGs Concatenados - 128 Características	0.0107	0.0105
TUGs Concatenados- 250 Características	0.0065	0.0057
TUGs Concatenados - 500 Características	0.0061	0.0050
Todo Sinal (Camadas Densas) - 80 Características	0.0250	0.0262

Aplicação de Algoritmos de Classificação

- Com os 7 conjuntos de características extraídas dos espectrogramas, foram implementados classificadores para discriminação dos pacientes entre perfis de caidores e não-caidores a partir das características extraídas
- Utilizando a biblioteca scikit-learn, foram implementados os seguintes algoritmos:
 - Random Forest
 - SVM
 - Ada Boosting
 - Gradient Boosting

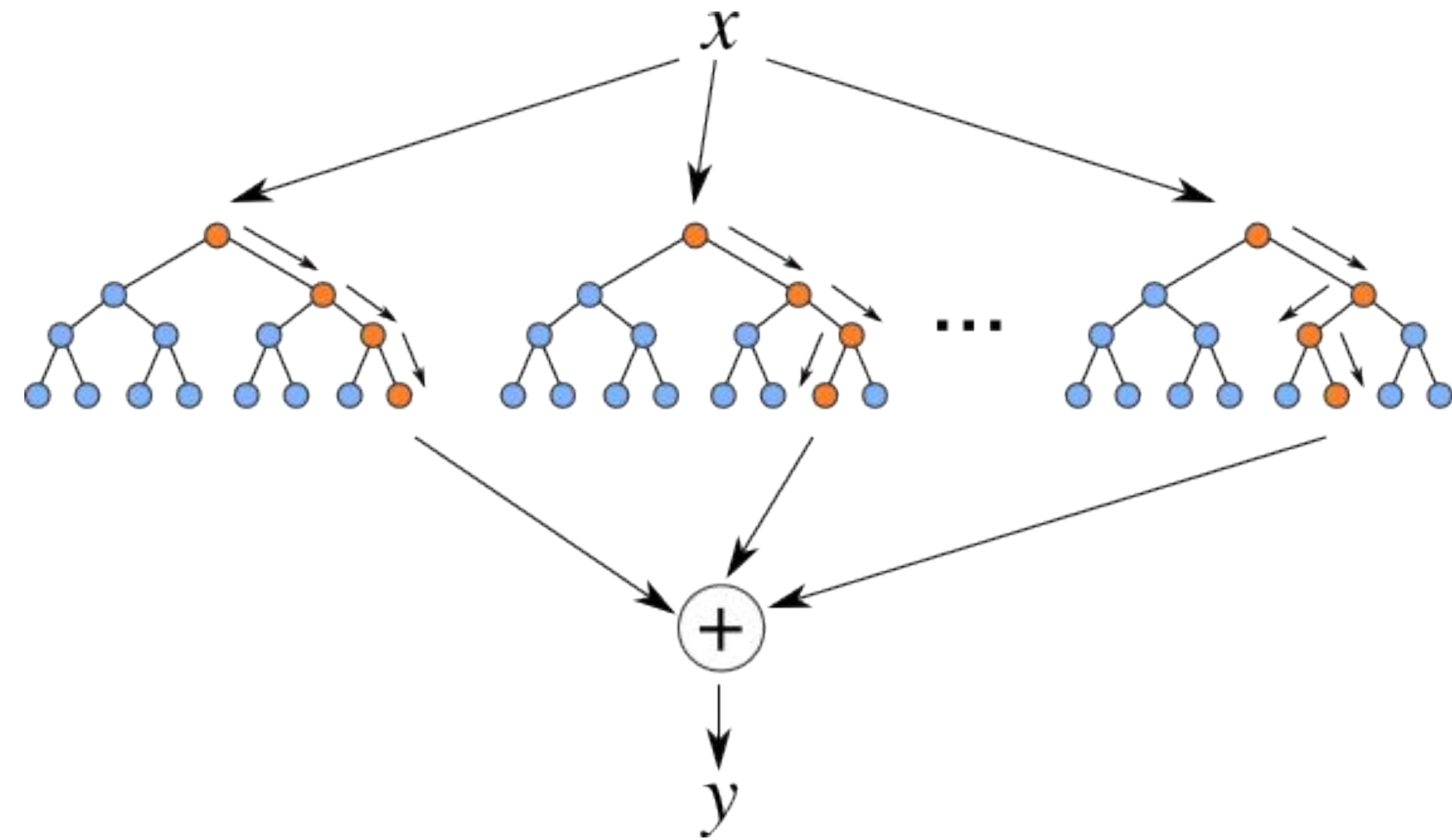
SVM

- Classificador que busca encontrar um hiperplano que melhor separa as classes de um problema
- A complexidade do hiperplano é definida pelo kernel utilizado, permitindo que o método permaneça útil em problemas não triviais



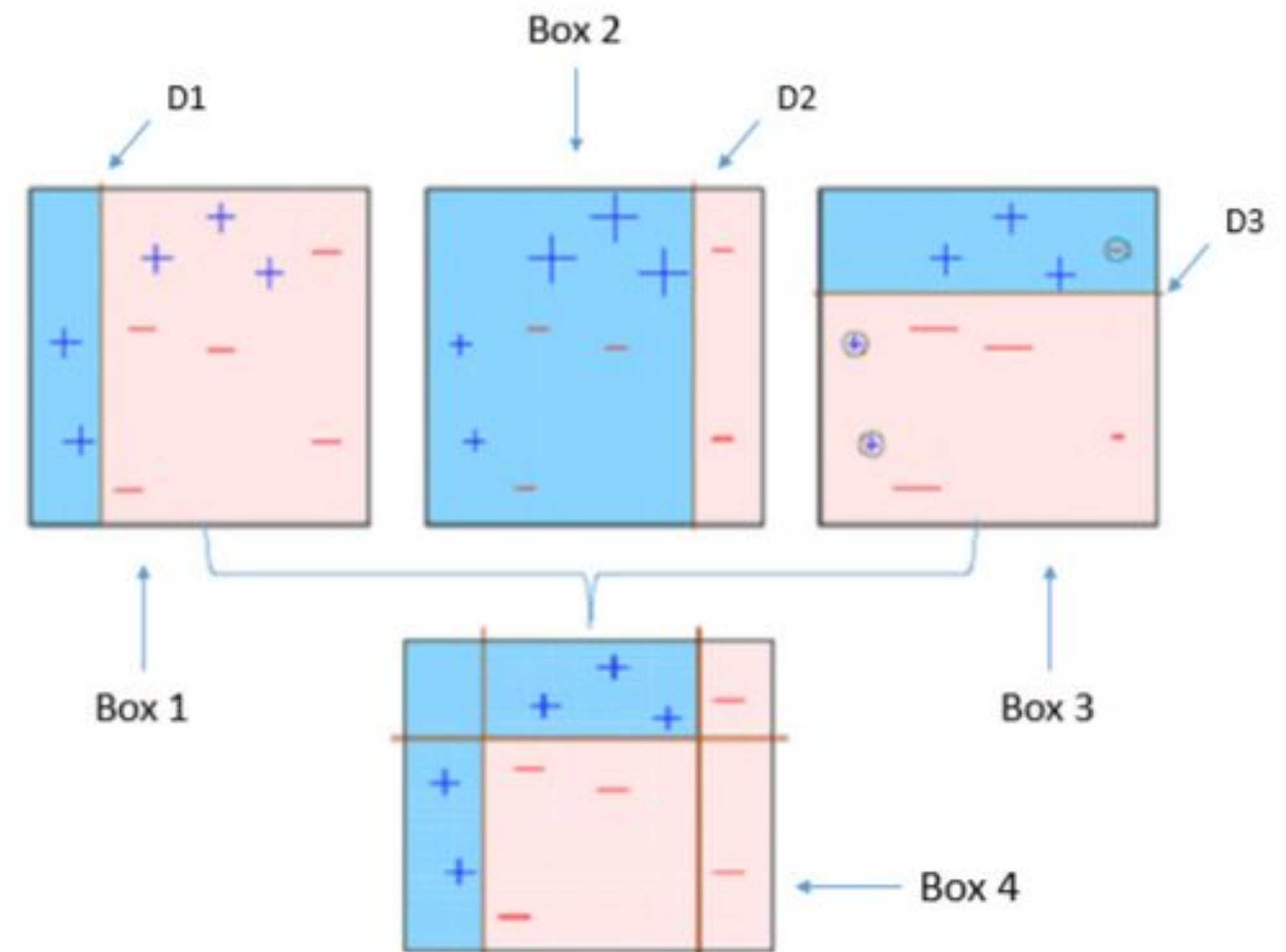
Random Forest

- Classificador baseado em *ensembles*
- Utiliza técnica chamada de *Bootstrap Aggregating*, ou *Bagging*
- Cria inúmeras árvores de decisão e utiliza um sistema de voto para determinar o resultado da predição



Ada Boosting e Gradient Boosting

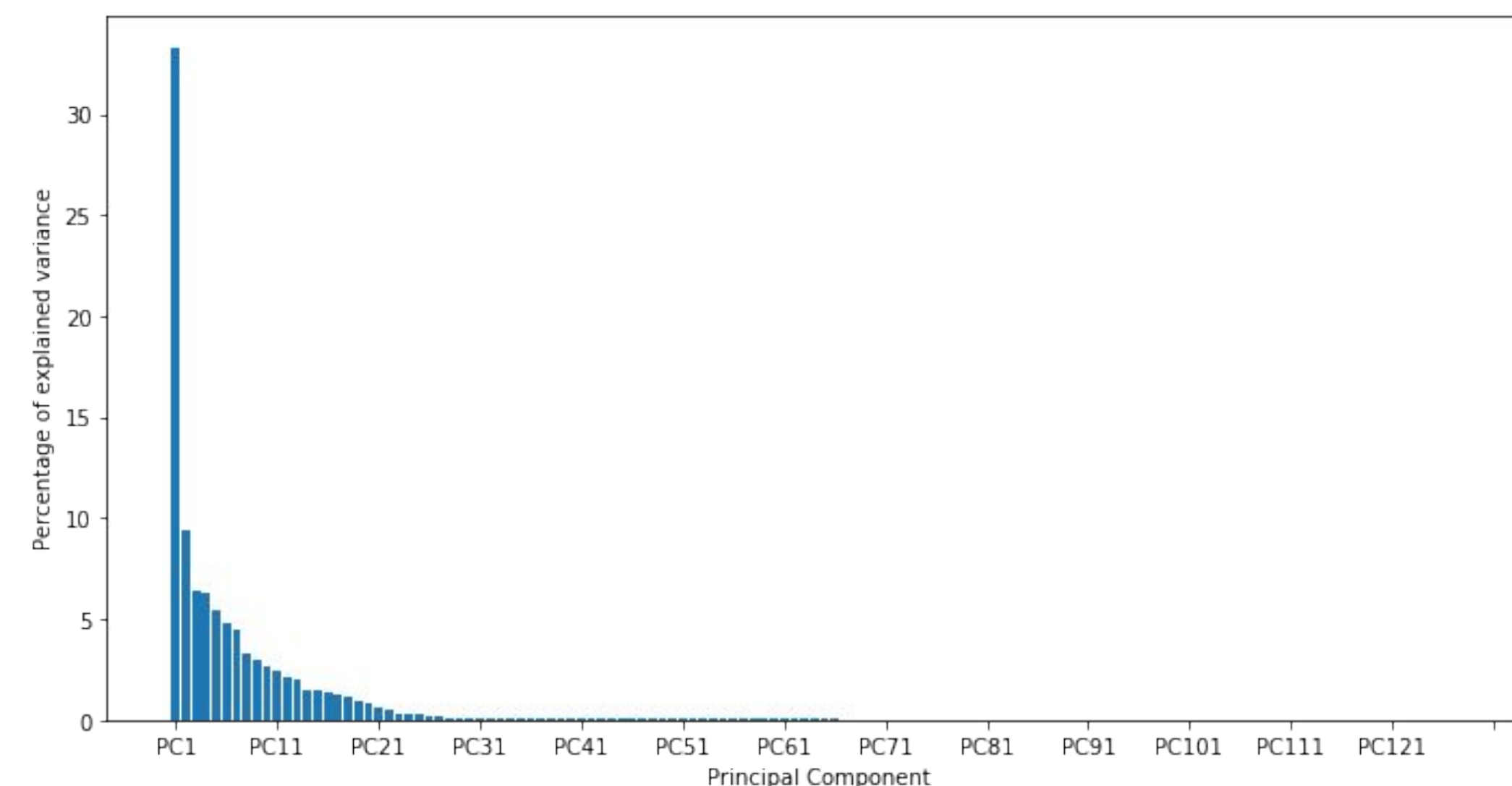
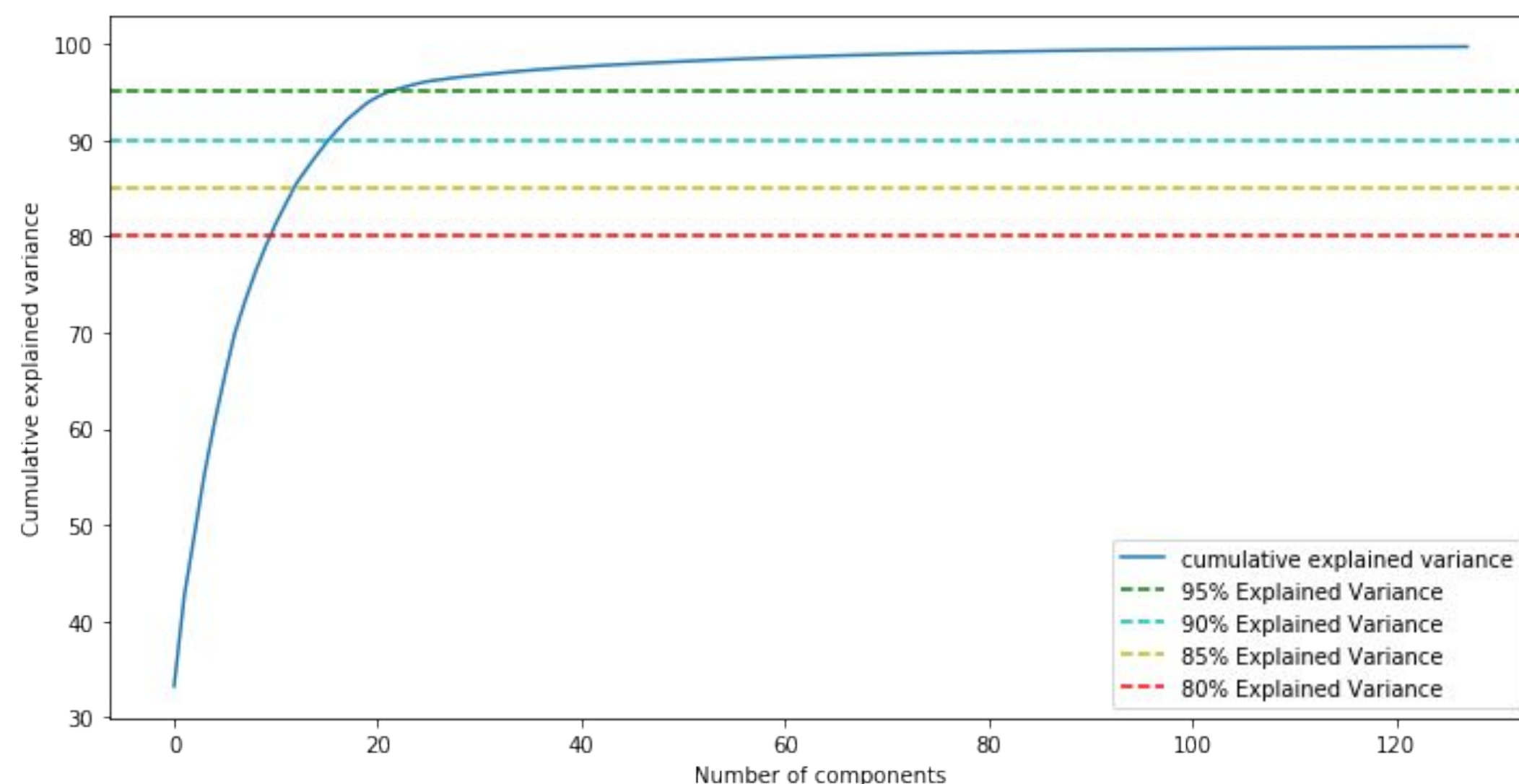
- Classificadores baseados em *ensembles*
- Utilizam técnica chamada de *Boosting*
- Criam um único classificador robusto através de um processo iterativo e sequencial de correção dos erros e reajuste dos pesos
- A diferença entre os métodos está em como o reajuste dos pesos é feito



Aplicação de Algoritmos de Classificação

- Para os algoritmos de SVM, Ada Boosting e Gradient Boosting, aplicou-se também a técnica de PCA (Principal Component Analysis)
 - Em cada um dos 3 algoritmos, aplicou-se o PCA selecionando 16, 32 e 64 principais componentes do conjunto de *features* extraídas

Scree Plot



Aplicação de Algoritmos de Classificação

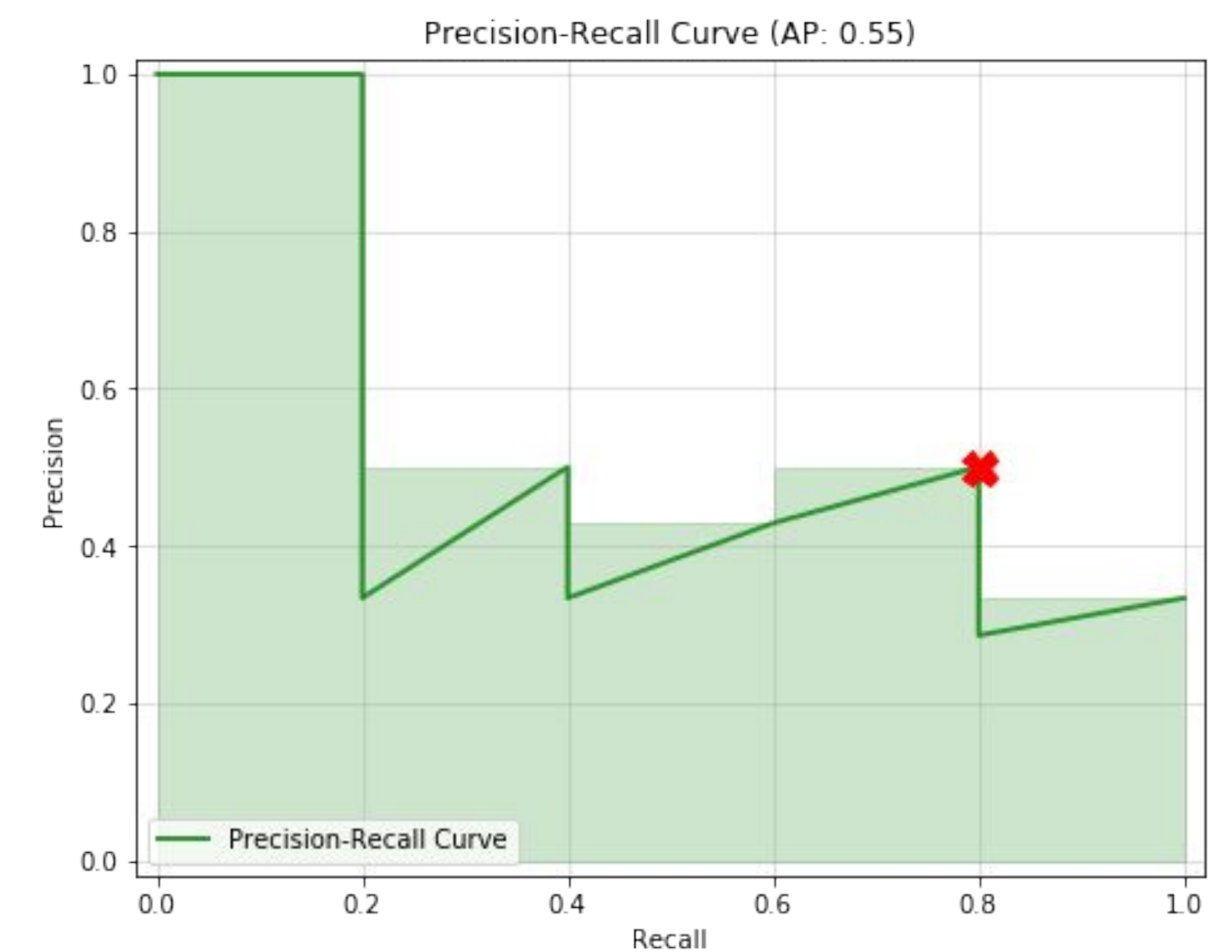
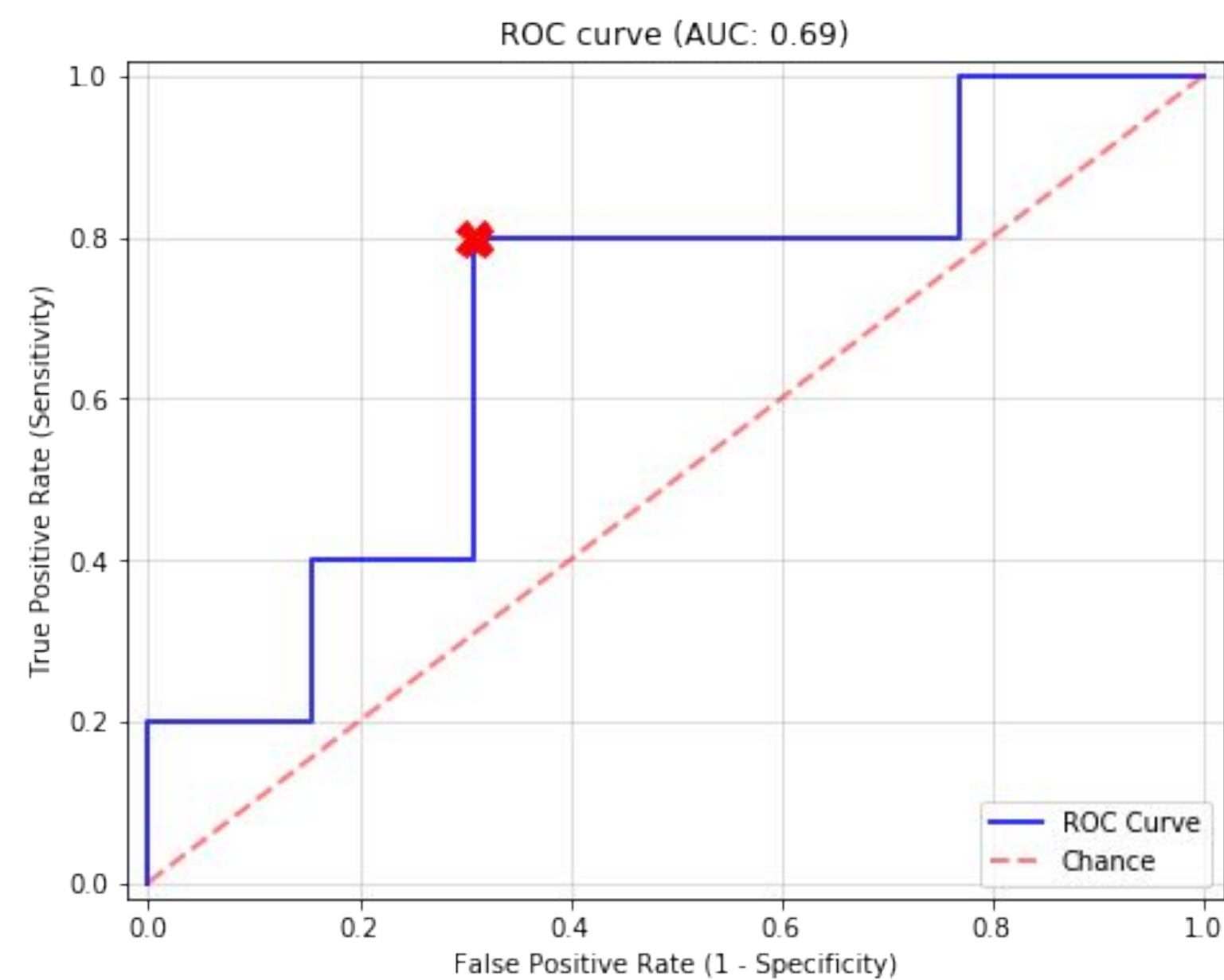
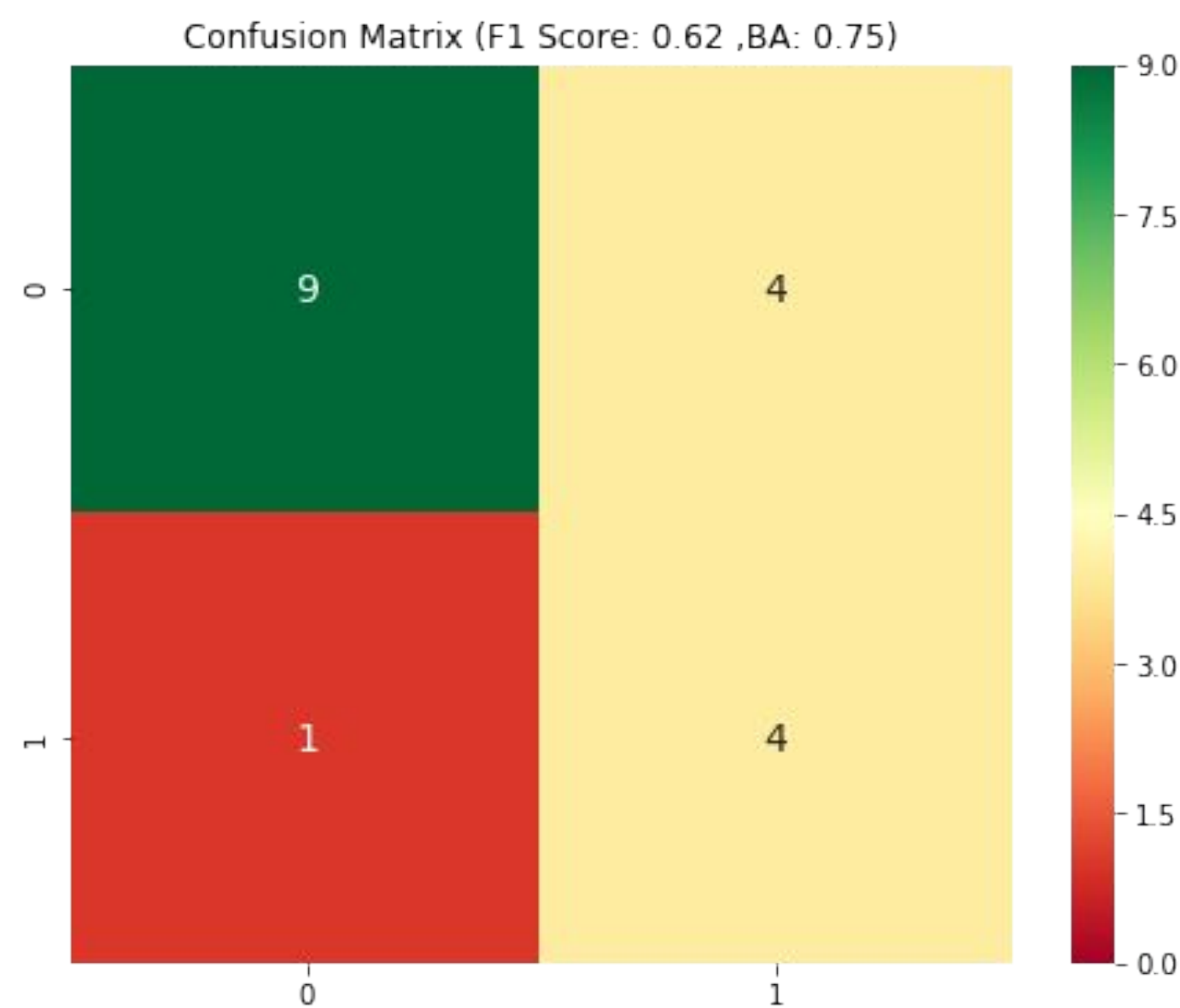
- Ao total, temos 10 diferentes classificadores para cada um dos 7 encoders treinados para extração de características, totalizando 70 modelos
- Foram criados também 4 modelos utilizando as características obtidas a partir dos dados de aceleração em frequência
- Logo, temos ao total 74 modelos para coletar resultados e avaliar

A decorative graphic on the left side of the slide, consisting of several overlapping teal-colored triangles and polygons of varying shades, creating a modern, abstract shape.

Resultados Obtidos

Resultados

- Para cada um dos 74 modelos criados, foram coletadas métricas de predição como as representadas as abaixo

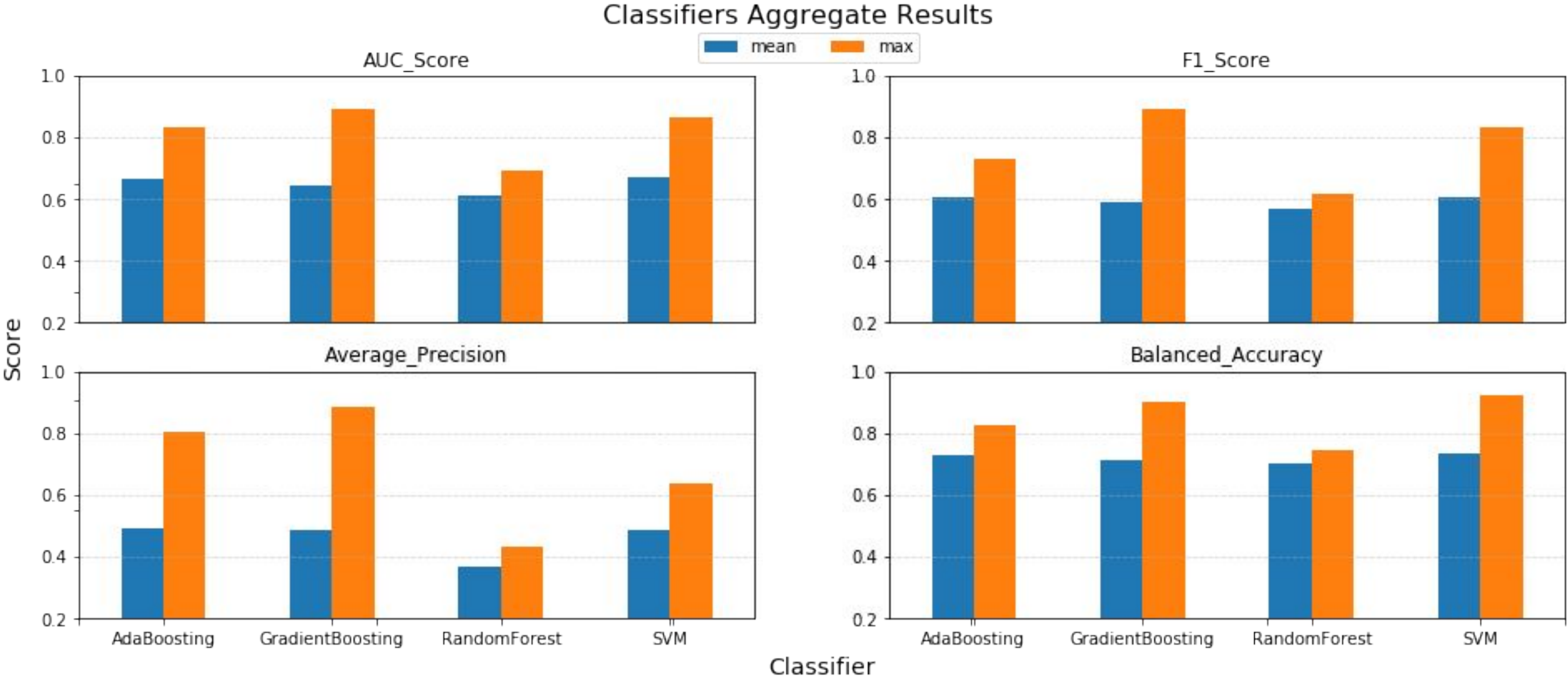
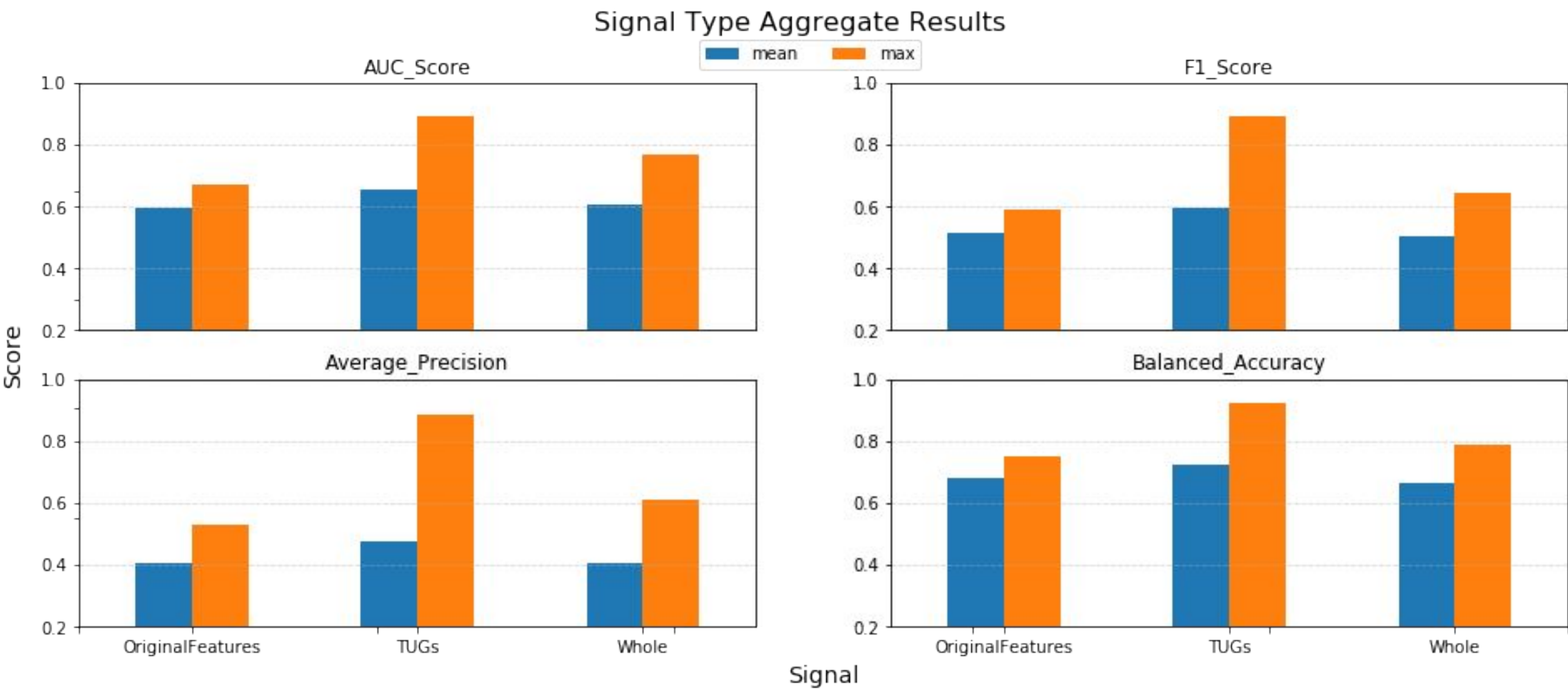


Resultados

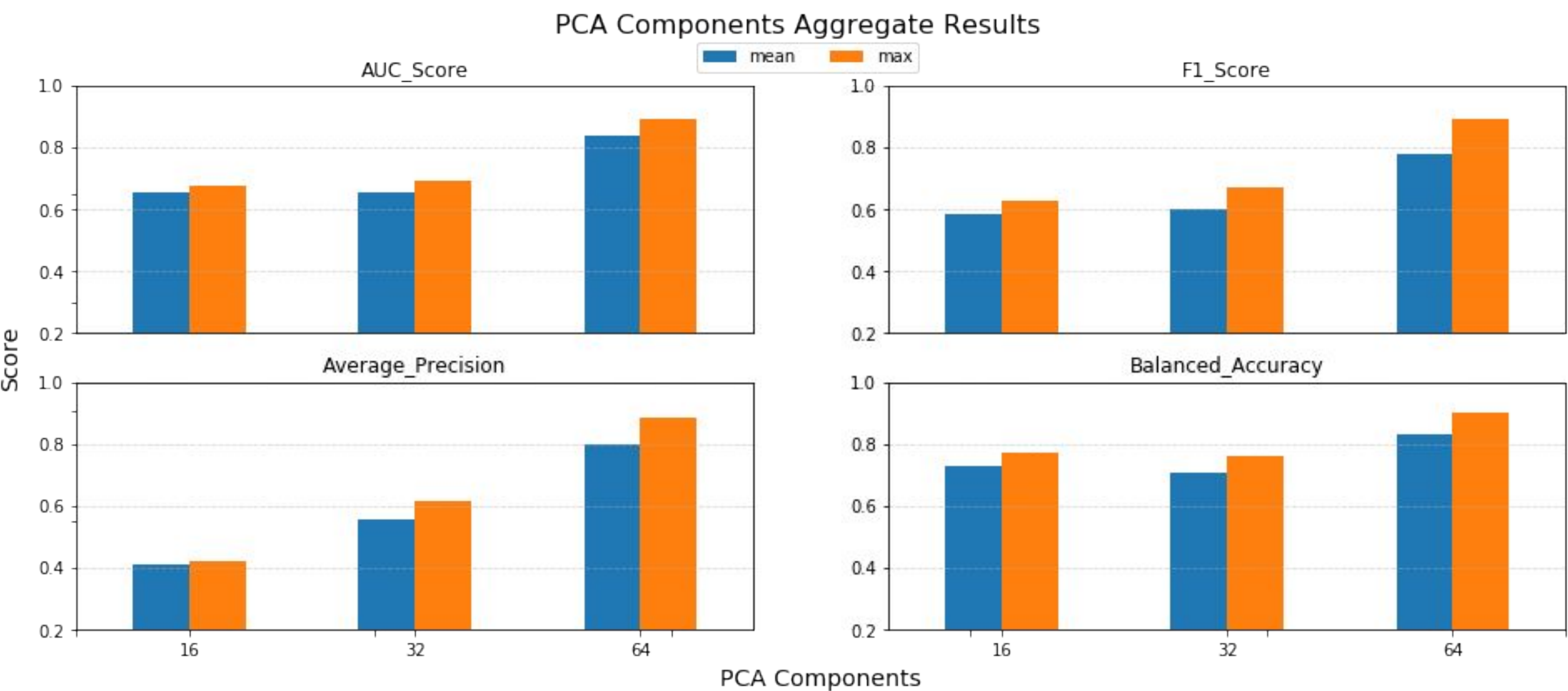
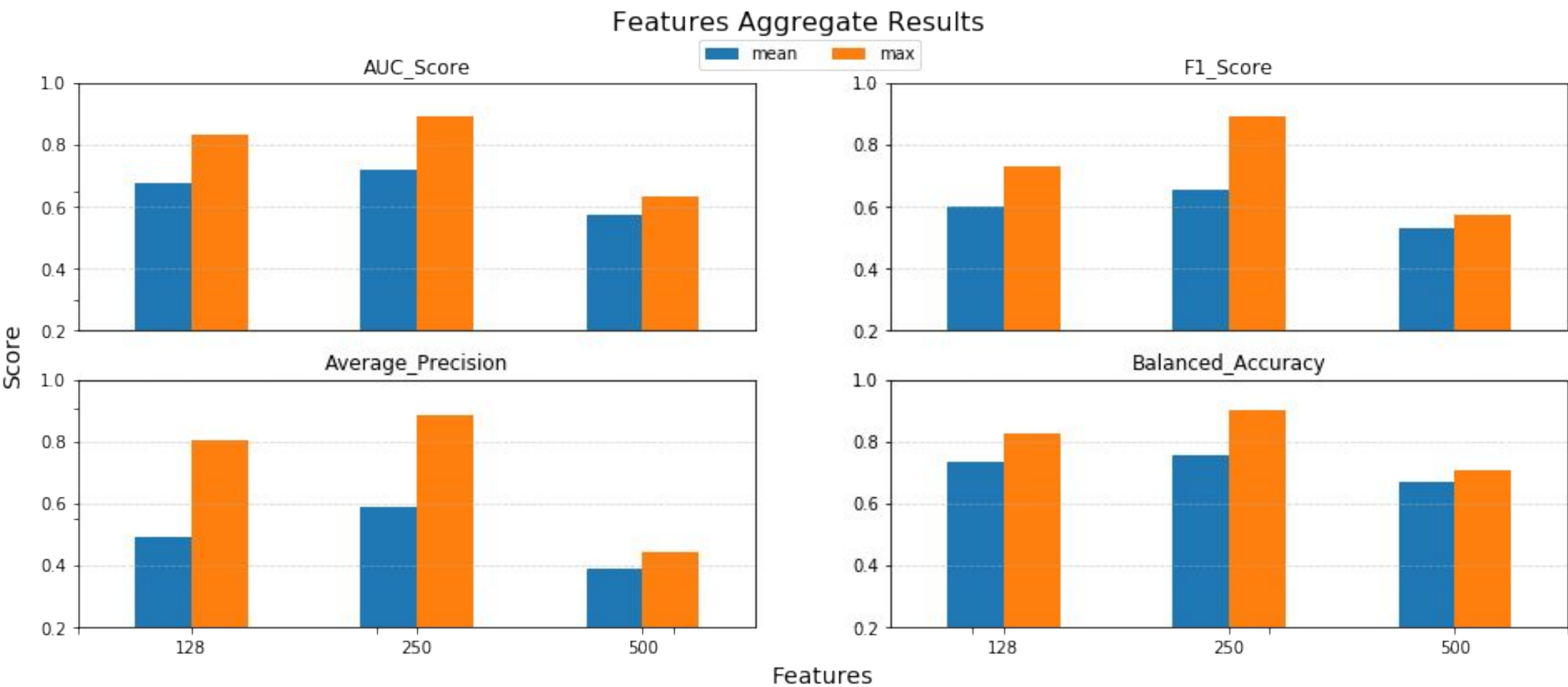
- Os 10 melhores modelos de acordo com a métrica AUC_Score

	AUC_Score	F1_Score	Average_Precision	Balanced_Accuracy
GradBoosting_TUGs_250_64	0.892	0.889	0.883	0.9
SVM_TUGs_500_32	0.862	0.833	0.596	0.923
AdaBoosting_TUGs_128_32	0.831	0.727	0.805	0.823
SVM_TUGs_128_64	0.8	0.667	0.633	0.808
AdaBoosting_TUGs_250_64	0.785	0.667	0.716	0.762
GradBoosting_272_16	0.768	0.643	0.526	0.789
AdaBoosting_TUGs_128_16	0.754	0.625	0.491	0.769
GradBoosting_272_32	0.743	0.581	0.553	0.736
AdaBoosting_272_16	0.739	0.571	0.524	0.721
SVM_TUGs_128_16	0.738	0.625	0.566	0.769

Resultados

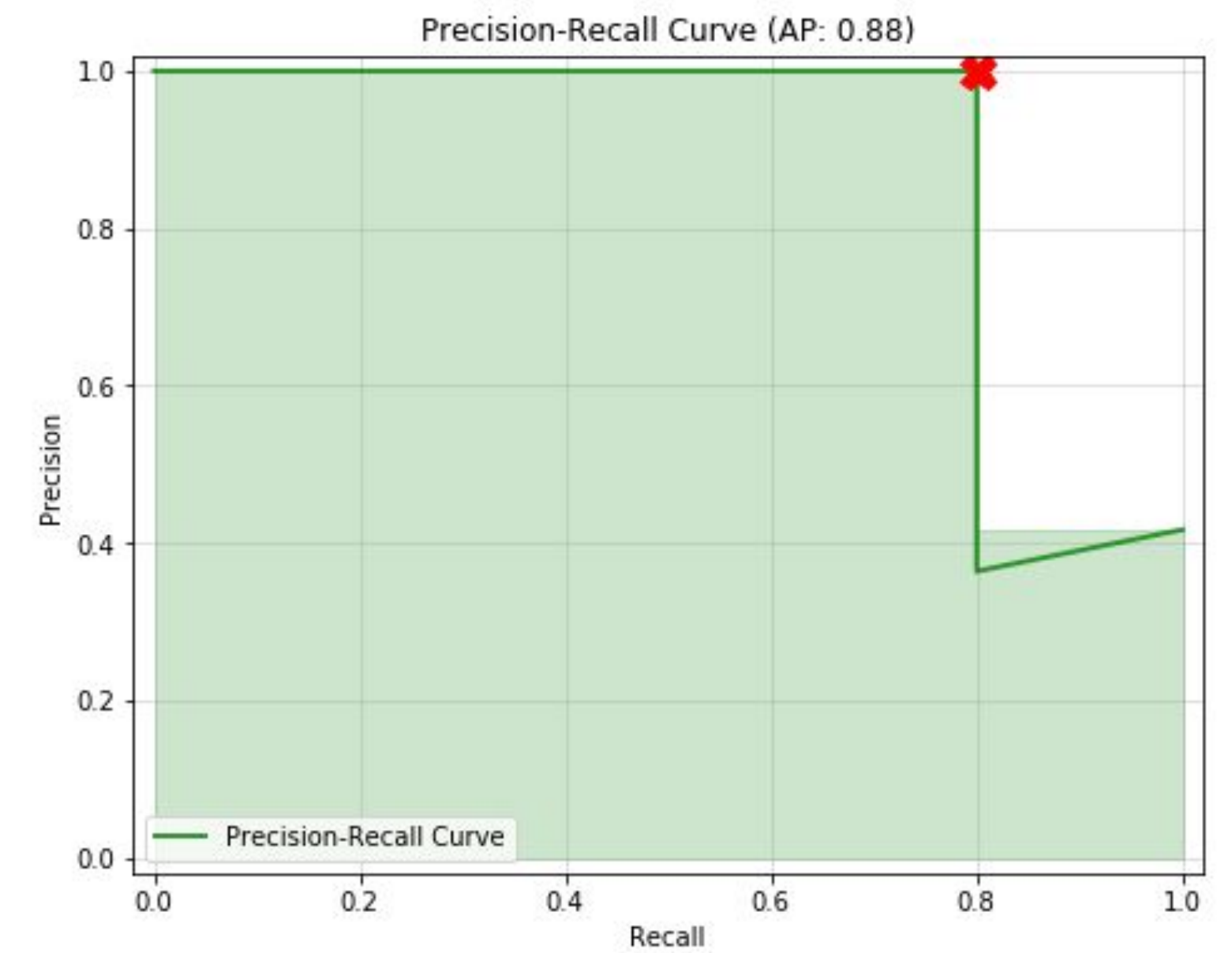
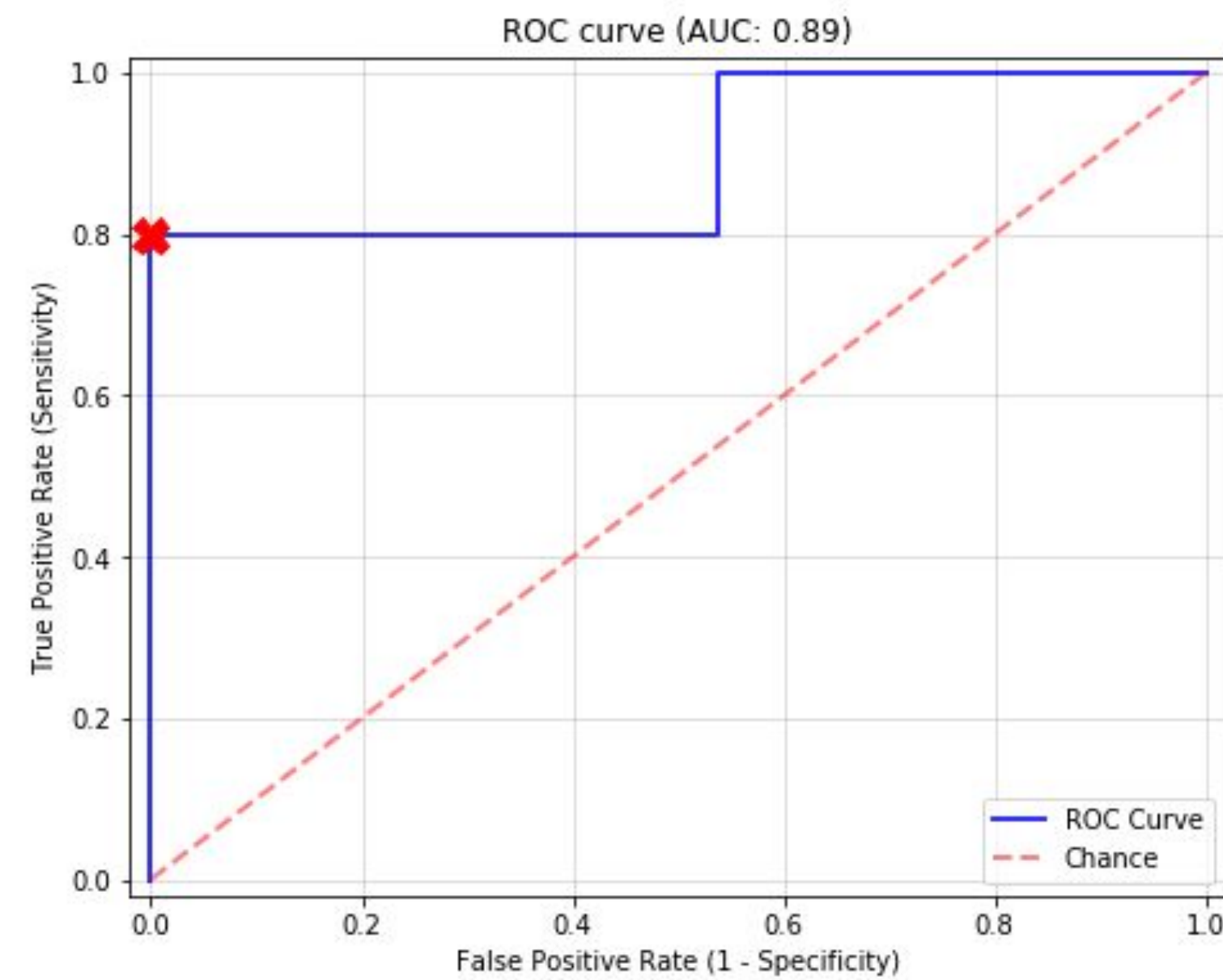
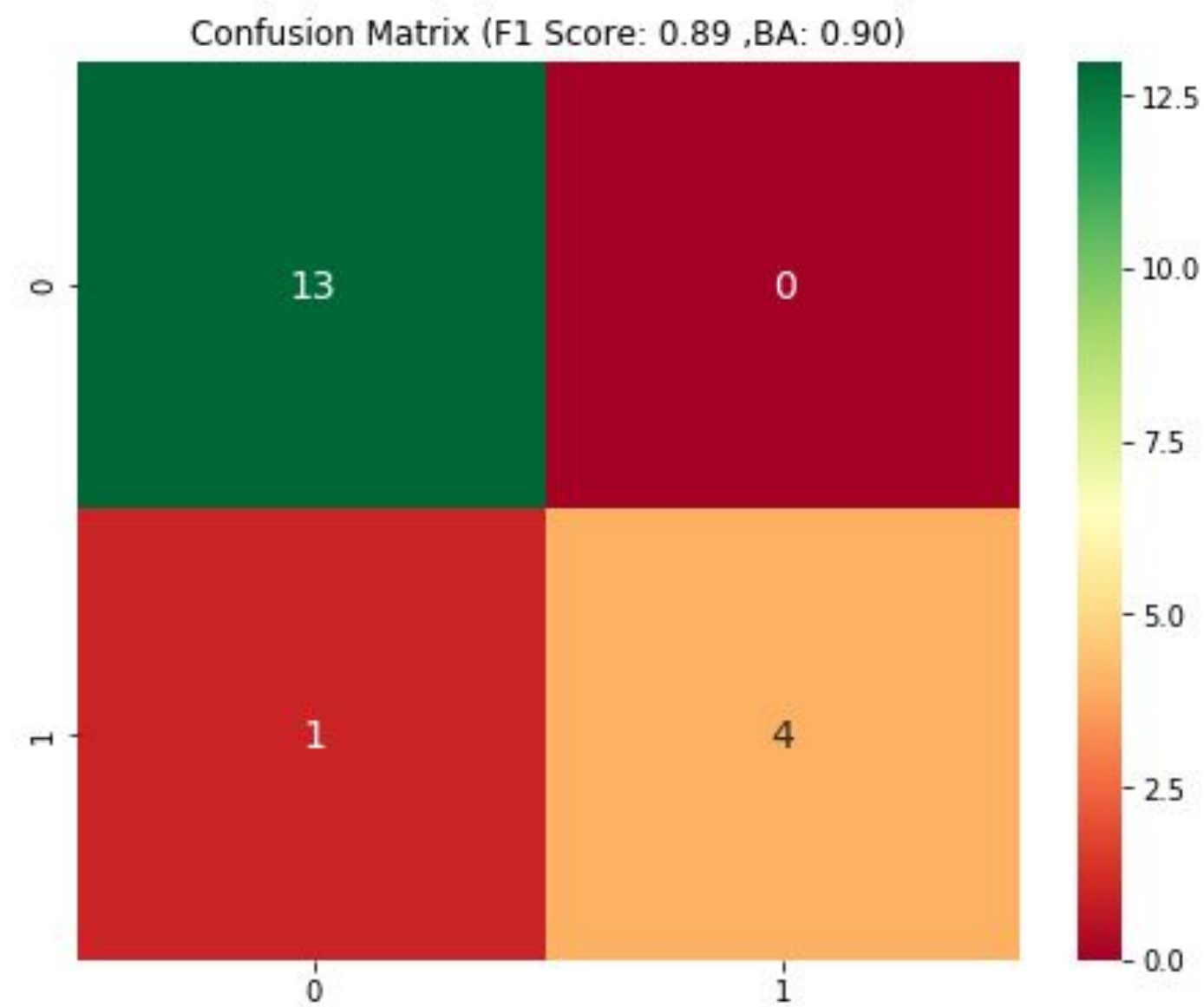


Resultados



Resultados

- O melhor modelo apresentou os seguintes resultados:



A decorative graphic on the left side of the slide, consisting of several overlapping teal-colored triangles and polygons of varying shades, creating a modern, abstract shape.

Conclusão

Aplicação de Algoritmos de Classificação

- Analisando as métricas obtidas e agregando-as por categorias, foi possível identificar quais conjuntos de dados e técnicas apresentaram os melhores resultados
- Dentre o total dos 74 modelos avaliados, os melhores apresentaram desempenho excepcional, aproximando-se do modelo perfeito, enquanto os piores se aproximam de um classificador aleatório
- No entanto, devido a amostra pequena da base de dados utilizada e pequena generalização dos modelos, o desvio padrão dos resultados é alto
- Os resultados obtidos, embora inconclusivos, demonstram bom potencial da metodologia empregada, podendo servir como base para trabalhos futuros



Obrigado!

Dúvidas?

