REDUÇÃO DE DIMENSIONALIDADE, RANQUEAMENTO E PREDIÇÃO DE CPUs E GPUs COM CIÊNCIA DE DADOS E MACHINE LEARNING

Davi Coene Rosa, Eduardo Peres Mosena, Tiago Rafael Wengrat e Thiago Schaedler Uhlmann*

RESUMO

Este artigo explora a aplicação prática da Ciência de Dados na análise comparativa de dados técnicos sobre unidades de processamento gráfico (GPUs) e unidades centrais de processamento (CPUs). Utilizando técnicas de manipulação, visualização e extração de conhecimento a partir de um banco de dados especializado, busca-se evidenciar padrões de desempenho, consumo e arquitetura entre esses dois componentes fundamentais da computação moderna, e descobrir o melhor processador e placa de vídeo nessa base de dados. O estudo fundamenta-se em princípios estatísticos e técnicas de análise exploratória de dados. A proposta visa não apenas ilustrar o potencial da Ciência de Dados em contextos técnicos, mas também contribuir com insights para decisões em áreas como engenharia de software, arquitetura de sistemas e inteligência artificial.

Palavras-chave: Ciência de Dados, GPU, CPU, Banco de Dados, Machine Learning.

DIMENSIONALITY REDUCTION, RANKING AND PREDICTION OF CPUs AND GPUs WITH DATA SCIENCE AND MACHINE LEARNING

ABSTRACT

This article explores the practical application of Data Science in the comparative analysis of technical data on Graphics Processing Units (GPUs) and Central Processing Units (CPUs). By employing techniques of data manipulation, visualization, and knowledge extraction from a specialized database, the study aims to highlight performance, power consumption, and architectural patterns between these two fundamental components of modern computing, and to identify the best processor and graphics card within the dataset. The study is grounded in statistical principles and exploration data analysis techniques. This approach not only illustrates the potential of Data Science in technical contexts but also provides valuable insights for decision-making in fields such as software engineering, system architecture, and artificial intelligence.

Key words: Data Science, GPU, CPU, Database, Machine Learning.

* Autor correspondente (e-mail): thiago.uhlmann@sistemafiep.org.br

1. INTRODUÇÃO

A Ciência de Dados tem se consolidado como um dos campos mais relevantes da era digital, promovendo *insights* que impactam desde decisões empresariais até avanços na pesquisa científica. No contexto tecnológico, um dos maiores focos de interesse reside na análise de hardware computacional, especialmente das unidades centrais de processamento (CPU) e das unidades de processamento gráfico (GPU), que constituem a espinha dorsal de qualquer sistema computacional de alto desempenho. A crescente demanda por sistemas eficientes, principalmente em áreas como computação científica, jogos, aprendizado de máquina e simulações em tempo real, torna indispensável uma compreensão aprofundada do desempenho e da evolução desses componentes. De acordo com Provost e Fawcett (2013), “a Ciência de Dados é a extração de conhecimento a partir de grandes volumes de dados por meio de análise, aprendizado e inferência”. Sob esse prisma, a presente pesquisa propõe a aplicação dessa ciência à avaliação de métricas técnicas de GPUs e CPUs, a partir de um banco de dados estruturado.

O objetivo deste artigo é descobrir entre as variáveis quais são os melhores GPUS e CPUS dentro dessa base de dados e apresentar uma análise orientada por dados que permita compreender, por meio de estatísticas descritivas e visualizações, como diferentes modelos de GPUs e CPUs se comportam em aspectos como frequência de clock, número de núcleos, TDP (*Thermal Design Power*), processos de fabricação e benchmarks de desempenho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Ciência de Dados

A Ciência de Dados é um campo interdisciplinar que une estatística, aprendizado de máquina, mineração de dados e computação para processar e interpretar grandes volumes de informação. Segundo Dhar (2013), “*Data Science* é a

nova forma de ciências empíricas, pois trabalha com grandes conjuntos de dados para modelar fenômenos complexos, com base em padrões observáveis". Seu valor reside na capacidade de transformar dados brutos em inteligência aplicável, muitas vezes por meio de *pipelines* automatizados de análise.

2.2 Estatística

A Estatística é uma ciência que coleta, organiza, analisa e interpreta dados com o objetivo de apoiar a tomada de decisões em condições de incerteza. Seu papel é essencial em diversas áreas do conhecimento, como economia, biologia, engenharia, medicina, ciências sociais, entre outras (TRIOLA, 2019). A Estatística pode ser dividida em dois grandes ramos: a Estatística Descritiva, que resume e descreve as principais características de um conjunto de dados, e a Estatística Inferencial, que utiliza uma amostra para tirar conclusões sobre uma população.

Os principais conceitos da Estatística envolvem medidas de tendência central (média, mediana e moda), medidas de dispersão (desvio padrão, variância e amplitude) e distribuição de probabilidades. Além disso, a aplicação de testes de hipóteses e modelos de regressão são amplamente utilizados em pesquisas quantitativas, oferecendo embasamento científico para validação de hipóteses e previsão de comportamentos (MONTGOMERY; RUNGER, 2020).

A utilização da Estatística vem crescendo exponencialmente com a era dos dados. Em um cenário dominado por *big data* e análise preditiva, o conhecimento estatístico tornou-se uma ferramenta indispensável para cientistas de dados, gestores e pesquisadores, contribuindo significativamente para a extração de valor e geração de conhecimento a partir de grandes volumes de dados.

2.3 Linguagem de programação

Linguagens de programação são conjuntos estruturados de regras e símbolos que permitem a comunicação entre seres humanos e computadores, facilitando o desenvolvimento de algoritmos e a construção de sistemas computacionais. Elas permitem que instruções sejam dadas ao computador de forma lógica e

compreensível, transformando ideias humanas em operações automatizadas (SEBESTA, 2012). As linguagens são classificadas em diferentes paradigmas, como imperativo, funcional, orientado a objetos e lógico, cada um com abordagens distintas de resolução de problemas computacionais.

A escolha da linguagem de programação depende de diversos fatores, como desempenho, facilidade de uso, aplicação desejada e suporte da comunidade. Por exemplo, linguagens como C e C++ são amplamente utilizadas em sistemas embarcados e aplicações que exigem alto desempenho, enquanto *Python* e *JavaScript* são preferidas em aplicações *web* e ciência de dados por sua sintaxe simples e ampla variedade de bibliotecas (TANENBAUM; BOS, 2015). Independentemente da linguagem escolhida, o domínio desses sistemas simbólicos é essencial para o desenvolvimento de soluções tecnológicas eficazes, sendo uma competência fundamental para profissionais da computação.

2.4 Ciências de Dados

A Ciência de Dados é uma área interdisciplinar que combina estatística, ciência da computação e conhecimento de domínio para extrair *insights* e conhecimento útil a partir de grandes volumes de dados. Seu objetivo principal é transformar dados brutos em informações valiosas para tomada de decisões, utilizando técnicas de análise exploratória, modelagem preditiva, visualização e aprendizado de máquina (PROVOST; FAWCETT, 2013).

Os profissionais da área — conhecidos como cientistas de dados — utilizam linguagens como *Python* e R, bancos de dados relacionais e não relacionais, além de ferramentas de *big data* e bibliotecas de *machine learning* para manipular, processar e interpretar conjuntos de dados cada vez mais complexos. A Ciência de Dados tornou-se fundamental em setores como saúde, finanças, marketing, segurança e tecnologia, sendo considerada uma das profissões mais promissoras do século XXI (MARR, 2016).

Além da parte técnica, a Ciência de Dados envolve forte pensamento crítico, curiosidade investigativa e capacidade de comunicação, pois os dados, por si só, não

possuem valor se não forem contextualizados e compreendidos de forma acessível pelos tomadores de decisão.

2.5 Bancos de Dados

Bancos de dados são estruturas organizadas para armazenamento, gerenciamento e recuperação de informações. Em análises modernas, bancos do tipo CSV (*Comma-Separated Values)* ou *SQL* são amplamente utilizados em tarefas de Ciência de Dados. De acordo com Elmasri e Navathe (2015), “a modelagem eficiente e a integridade dos dados são fatores críticos para análises confiáveis e extração de conhecimento real”. No caso do presente trabalho, a análise será feita sobre um *dataset* técnico contendo especificações de diferentes modelos de CPUs e GPUs. A manipulação desses dados exige tratamento de inconsistências, normalização e seleção de variáveis relevantes — tarefas típicas da fase de pré-processamento em *Data Science* (Han, Pei & Kamber, 2011).

2.6 CPU e GPU: Arquitetura e Aplicações

A CPU (*Central Processing Unit*) é projetada para lidar com tarefas sequenciais e decisões lógicas rápidas, enquanto a GPU (*Graphics Processing Unit*) se destaca por seu poder de processamento paralelo massivo, fundamental para tarefas de computação gráfica e algoritmos de inteligência artificial. Como pontua Hennessy e Patterson (2011), “GPUs representam a maior evolução arquitetônica da computação paralela nas últimas duas décadas”. Ambos os componentes evoluíram significativamente, refletindo-se em diferentes métricas: litografia (nm), número de threads, caches, largura de banda e consumo térmico. A análise desses dados permite observar tendências de desenvolvimento, relação custo-benefício e limites físicos da microarquitetura.

2.7 Linguagem R

A linguagem R é um ambiente e linguagem de programação voltado principalmente para análise estatística, mineração de dados e gráficos. Criada por Ross Ihaka e Robert Gentleman na década de 1990, R se destaca por ser *open source* e por sua grande comunidade acadêmica e científica. É amplamente utilizada em estatística aplicada, bioinformática, econometria, aprendizado de máquina e ciência de dados (R CORE TEAM, 2024).

Uma das maiores vantagens do R é sua vasta coleção de pacotes disponíveis através do CRAN (*Comprehensive R Archive Network*), que permite aos usuários implementar desde análises estatísticas básicas até modelos complexos de *machine learning*. Além disso, o R oferece ferramentas poderosas de visualização gráfica, como os pacotes ggplot2 e lattice, que possibilitam a criação de gráficos altamente customizáveis e de alta qualidade (WICKHAM; GROLEMUND, 2017).

Do ponto de vista acadêmico e científico, R é valorizado por sua reprodutibilidade, capacidade de integração com outras linguagens (como Python e C++) e forte apoio da comunidade científica. Seu uso vem crescendo também em ambientes corporativos devido à sua robustez na manipulação de grandes volumes de dados e sua aplicabilidade em áreas como finanças, saúde e marketing (R CORE TEAM, 2024; WICKHAM; GROLEMUND, 2017).

2.8 *Machine Learning*

O *Machine Learning* (Aprendizado de Máquina) é uma subárea da inteligência artificial que estuda algoritmos capazes de aprender padrões a partir de dados, sem a necessidade de programação explícita para cada tarefa. Essa abordagem permite que os sistemas adaptem seu comportamento com base em experiências anteriores, o que tem impulsionado sua aplicação em áreas como diagnósticos médicos, previsões financeiras, reconhecimento de padrões e automação industrial (ALPAYDIN, 2020).

Os algoritmos de aprendizado de máquina podem ser classificados em três grandes categorias: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. No aprendizado supervisionado, os modelos são treinados com dados rotulados e buscam prever um resultado com base em exemplos anteriores. Já o aprendizado não supervisionado, busca descobrir padrões ocultos em conjuntos de dados sem rótulos. O aprendizado por reforço, por sua vez, envolve a tomada de decisões por agentes autônomos que aprendem a partir de recompensas e penalidades recebidas em um ambiente (GOODFELLOW; BENGIO; COURVILLE, 2016).

O crescimento no volume de dados e o aumento da capacidade computacional têm favorecido o uso do Machine Learning em soluções cada vez mais complexas e escaláveis, tornando-se um pilar fundamental da transformação digital nas organizações.

2.9 Análise descritiva

A análise descritiva é uma das etapas fundamentais da estatística e da ciência de dados, voltada para o resumo e interpretação de dados com o objetivo de entender suas principais características antes da aplicação de modelos preditivos ou inferenciais. Por meio de técnicas como medidas de tendência central (média, mediana e moda), medidas de dispersão (desvio padrão, variância e amplitude), tabelas de frequência e representações gráficas, a análise descritiva permite uma visualização clara do comportamento dos dados (TRIOLA, 2019).

Essa etapa é essencial para identificar padrões, outliers, tendências e possíveis erros nos dados, servindo como base para tomadas de decisão mais embasadas. Ferramentas como histogramas, *boxplots* e gráficos de dispersão são amplamente utilizadas para ilustrar os dados de maneira acessível, facilitando a comunicação entre analistas e gestores (MONTGOMERY; RUNGER, 2020).

Em contextos corporativos e acadêmicos, a análise descritiva é considerada o primeiro passo de qualquer processo analítico, pois fornece um diagnóstico inicial que

orienta as etapas seguintes da investigação, como a modelagem estatística ou o uso de algoritmos de *machine learning*.

2.10 Análise preditiva

A análise preditiva é uma técnica estatística e computacional que utiliza dados históricos, algoritmos de *machine learning* e modelos matemáticos para prever eventos futuros ou comportamentos prováveis. Trata-se de uma evolução da análise descritiva e inferencial, pois além de entender e explicar os dados, seu objetivo é antecipar resultados com base em padrões previamente identificados (SHMUELI; KOPPIUS, 2011).

Esse tipo de análise é amplamente utilizado em setores como finanças, *marketing*, saúde e logística para prever riscos, otimizar recursos, identificar fraudes e melhorar a tomada de decisões. Entre os métodos mais comuns estão regressão linear e logística, redes neurais artificiais, árvores de decisão, máquinas de vetor de suporte (SVM) e modelos baseados em séries temporais (JAMES et al., 2021).

A eficácia da análise preditiva depende da qualidade dos dados utilizados, do conhecimento do domínio de aplicação e da validação rigorosa dos modelos construídos. Quando bem implementada, ela permite às organizações obter vantagem competitiva ao antecipar cenários e agir de forma proativa.

2.11 *K-means*

O algoritmo *K-means* é uma técnica de aprendizado não supervisionado amplamente utilizada para agrupamento (*clustering*) de dados. Seu objetivo é particionar um conjunto de dados em K grupos distintos, de modo que os objetos dentro de um mesmo grupo sejam mais semelhantes entre si do que aos objetos dos outros grupos (MACQUEEN, 1967). Para isso, o algoritmo define K centroides iniciais e atribui cada ponto de dados ao centroide mais próximo, recalculando as posições dos centroides iterativamente até que a convergência seja alcançada.

K-means é valorizado por sua simplicidade, eficiência computacional e aplicabilidade em diversas áreas, como segmentação de clientes, análise de imagens

e bioinformática. Contudo, o algoritmo possui limitações, como a necessidade de pré-definir o número K de clusters e a sensibilidade a pontos fora da curva (*outliers*) e à escolha dos centroides iniciais, o que pode levar a soluções subótimas (JAIN, 2010).

Para mitigar essas limitações, variantes do algoritmo e métodos de validação de *clusters*, como o método do cotovelo (*elbow method*), são frequentemente utilizados para determinar o número ideal de grupos e melhorar a estabilidade do agrupamento.

2.12 Análise de Componentes Principais (PCA)

A Análise de Componentes Principais (PCA) é uma técnica estatística utilizada para reduzir a dimensionalidade de conjuntos de dados multidimensionais, mantendo a maior parte da variabilidade presente nos dados originais. O PCA transforma variáveis correlacionadas em um novo conjunto de variáveis ortogonais, chamadas componentes principais, que são ordenadas de acordo com a quantidade de variância que explicam (JOLLIFFE, 2002).

Essa redução dimensional facilita a visualização, o processamento e a modelagem de dados, especialmente quando se trabalha com grandes volumes de informações ou com muitas variáveis. O PCA é amplamente utilizado em áreas como reconhecimento de padrões, compressão de dados e análise exploratória, auxiliando na identificação de estruturas latentes nos dados (WICKHAM, 2017).

Embora eficiente, o PCA assume linearidade nas relações entre variáveis e que os componentes principais são combinações lineares das variáveis originais, o que pode limitar seu desempenho em dados altamente não lineares (JOLLIFFE, 2002).

3 ANÁLISE DESCRITIVA

A análise descritiva é uma etapa fundamental no processo de interpretação e exploração de conjuntos de dados. Seu principal objetivo é oferecer uma visão panorâmica sobre o comportamento das variáveis envolvidas, permitindo identificar padrões, inconsistências, tendências centrais e dispersões. No presente trabalho,

foram analisadas variáveis técnicas extraídas de um banco de dados contendo informações sobre unidades de processamento central (CPUs) e unidades de processamento gráfico (GPUs). Os dados analisados incluem consumo energético (TDP), tamanho do chip (*Die Size*), litografia, frequência de operação, número de transistores, entre outros.

Com a análise descritiva é possível retirar alguns tópicos muito importantes, uma delas é ter a visão geral de como se comporta a base de dados, como a da GPU, quanto a CPU, com isso foram realizados alguns cálculos estatísticos para observar como esses dados se comportam.

3.2 Categorização e Tabela de Frequências

Inicialmente, realizou-se a identificação e separação das variáveis do *dataset* em dois grupos principais: variáveis numéricas e variáveis categóricas. Entre as numéricas, destacam-se: TDP (W), *Die Size* (mm²), Freq. (MHz), *Transistors* (milhões) e *Process Size* (nm). As variáveis categóricas incluíram *Type*, *Vendor*, *Foundry* e *Product.*

Essa classificação inicial é essencial para a aplicação correta das técnicas estatísticas posteriores, uma vez que cada tipo de variável exige um tratamento distinto quanto à agregação, modelagem e visualização.

Figura 1 – Categorização dos dados

```
'data.frame':   4854 obs. of  14 variables:
 $ X                   : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Product             : chr  "AMD Athlon 64 3500+" "AMD Athlon 200GE" "Intel Core i5-1145G7" "Intel Xeon E5-2603 v2" ...
 $ Type                : chr  "CPU" "CPU" "CPU" "CPU" ...
 $ Release.Date        : chr  "2007-02-20" "2018-09-06" "2020-09-02" "2013-09-01" ...
 $ Process.Size..nm.   : num  65 14 10 22 45 22 65 65 10 90 ...
 $ TDP..W.             : num  45 35 28 80 125 95 125 130 28 89 ...
 $ Die.Size..mm.2.     : num  77 192 NA 160 258 160 285 140 NA 156 ...
 $ Transistors..million.: num  122 4800 NA 1400 758 1400 450 376 NA 154 ...
 $ Freq..MHz.          : num  2200 3200 2600 1800 3700 2400 2400 3000 2000 2200 ...
 $ Foundry             : chr  "Unknown" "Unknown" "Intel" "Intel" ...
 $ Vendor              : chr  "AMD" "AMD" "Intel" "Intel" ...
 $ FP16.GFLOPS         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ FP32.GFLOPS         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ FP64.GFLOPS         : num  NA NA NA NA NA NA NA NA NA NA ...
Colunas numéricas:
 X Process.Size..nm. TDP..W. Die.Size..mm.2. Transistors..million. Freq..MHz. FP16.GFLOPS FP32.GFLOPS FP64.GFLOPS
Colunas categóricas:
 Product Type Release.Date Foundry Vendor
```

Começando com a categorização dos dados, para definir os seus tipos, assim conseguindo escolher quais as variáveis melhores para trabalhar.

Em seguida, foi construída duas tabelas de frequências para a variável TDP, a primeira tabela é da CPU, enquanto a outra fala a cerca da GPU, e assim foi-se dividindo os dados em classes. Para definir os intervalos, utilizou-se a Regra de *Sturges*, que calcula o número ideal de classes com base na quantidade de observações:

$$\mathbf{k = 1 + 3.322 \cdot log\,10\,(n)}$$

Com base na regra, foram identificadas 13 classes distintas de TDP, como apresentado abaixo:

Figura 2 – Tabela de frequência das CPUs.

Classe	Frequencia	Frequencia_Acumulada	Frequencia_Relativa	Frequencia_Relativa_Acumulada
[1,31.7]	375	375	0.171	0.171
(31.7,62.4]	562	937	0.256	0.427
(62.4,93.1]	632	1569	0.288	0.716
(93.1,124]	269	1838	0.123	0.839
(124,154]	187	2025	0.085	0.924
(154,185]	68	2093	0.031	0.955
(185,216]	27	2120	0.012	0.967
(216,247]	27	2147	0.012	0.979
(247,277]	19	2166	0.009	0.988
(277,308]	22	2188	0.010	0.998
(308,339]	1	2189	0.000	0.999
(339,369]	1	2190	0.000	0.999
(369,400]	2	2192	0.001	1.000

Esse tipo de estrutura facilita a compreensão da distribuição dos dados e fornece suporte para análises subsequentes como histogramas, regressões e agrupamentos. Além disso, foram identificadas frequências relativas acumuladas, que permitem avaliar a proporção de valores abaixo de certos limiares técnicos. Por exemplo, 71.6% dos dispositivos possuem TDP inferior a 93,1W, valor que pode ser interpretado como limite superior típico para CPUs comuns.

Figura 3 – Tabela de frequência das GPUs.

Classe	Frequencia	Frequencia_Acumulada	Frequencia_Relativa	Frequencia_Relativa_Acumulada
[2,76.8]	1330	1330	0.653	0.653
(76.8,152]	356	1686	0.175	0.828
(152,226]	161	1847	0.079	0.907
(226,301]	146	1993	0.072	0.979
(301,376]	21	2014	0.010	0.989
(376,451]	4	2018	0.002	0.991
(451,526]	6	2024	0.003	0.994
(526,601]	4	2028	0.002	0.996
(601,676]	4	2032	0.002	0.998
(676,750]	0	2032	0.000	0.998
(750,825]	3	2035	0.001	1.000
(825,900]	1	2036	0.000	1.000

Agora com a tabela de frequência para as GPUS os valores de TDP são um pouco maiores, sendo cerca de 90.7% abaixo de 226W de consumo. A categorização e tabulação revelam que os dados apresentam assimetria positiva, com maior concentração em faixas energéticas inferiores. Essa constatação fundamenta as hipóteses verificadas posteriormente em modelos preditivos e regressões.

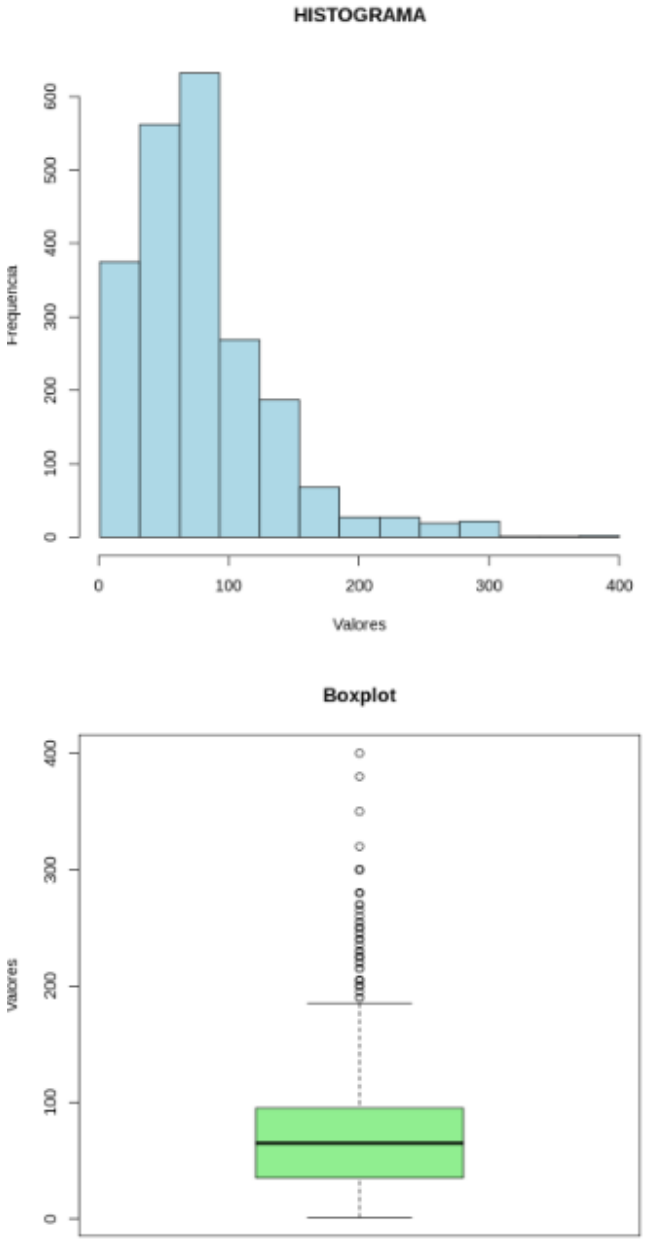
3.3 Histogramas e *Boxplots*

Com o objetivo de visualizar a distribuição das variáveis contínuas, especialmente o TDP (*Thermal Design Power*), foram elaborados histogramas e *boxplots* para as bases de CPU e GPU separadamente, além de uma visão geral da base completa. As visualizações permitem verificar padrões de dispersão, simetria e a presença de valores extremos, que nem sempre são detectáveis por estatísticas descritivas isoladas.

O histograma da CPU evidencia uma distribuição assimétrica à direita, com maior concentração de valores entre 10W e 150W. A frequência absoluta nas classes iniciais

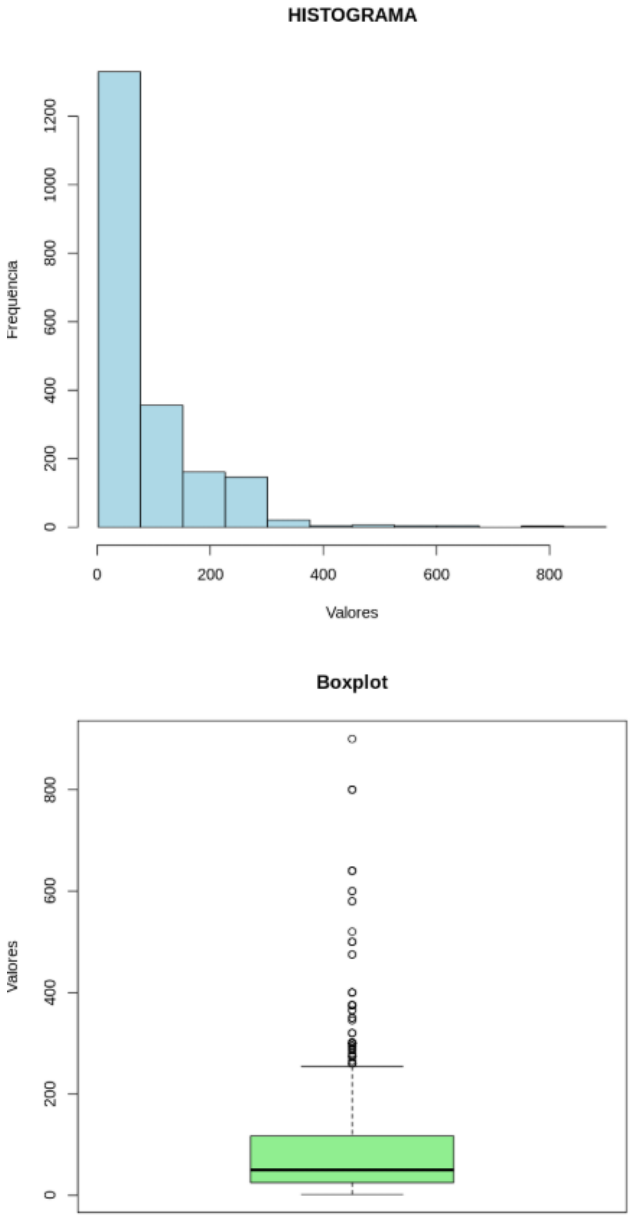
é significativamente maior, refletindo o comportamento padrão de dispositivos de consumo moderado. A queda abrupta nas classes superiores mostra que valores de TDP acima de 250W são exceções associadas a componentes de alto desempenho, geralmente GPUs.

Figura 4– Histograma e *Boxplot* da CPU



O *boxplot* revela uma mediana bem-posicionada próxima a 90W, além de uma faixa interquartil relativamente estreita. A presença de uma longa cauda superior com diversos outliers reforça a existência de componentes com consumo fora do padrão, especialmente CPUs de nova geração.

Figura 5– Histograma e *Boxplot* da GPU



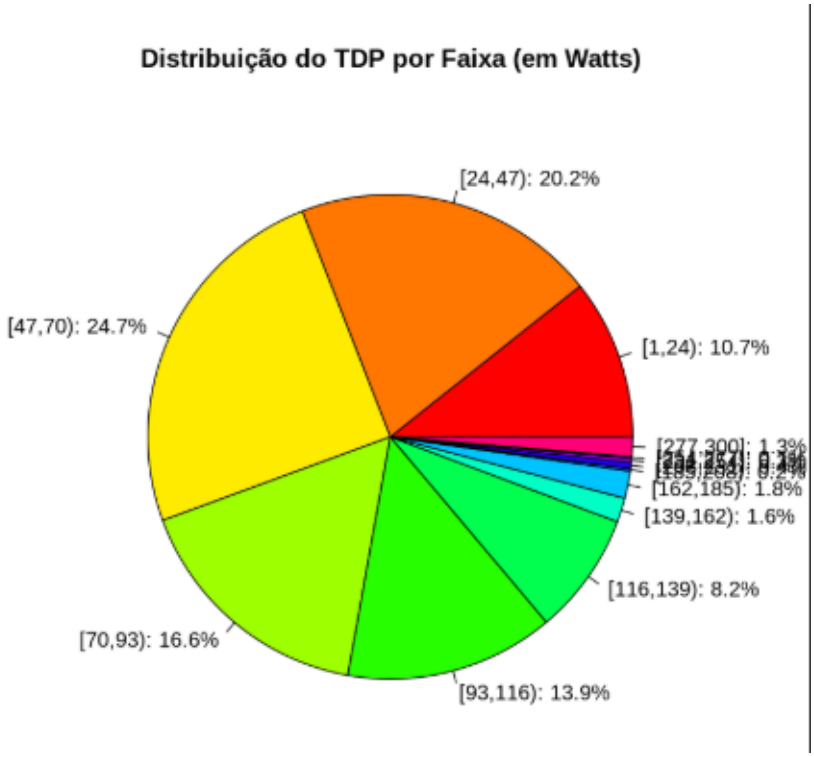
O histograma das GPUS tem um comportamento um pouco diferente, por mais que a faixa de consumo média esteja muito próximo as das CPUS, as anormalidades estão muito além da mediana e dá média, devido a isso o gráfico se estende muito mais à esquerda.

O *boxplot* representa muito bem isso, com anormalidades que passam de 800 W, que assim como os processadores devem representar GPUS de altíssimo desempenho.

3.4 Gráficos de Pizza e Gráficos de Colunas

Além da visualização do consumo por histogramas e *boxplot* também é possível observar alguns valores em gráfico de pizza, para os gráficos foram realizados valores para CPUS e GPUS respectivamente:

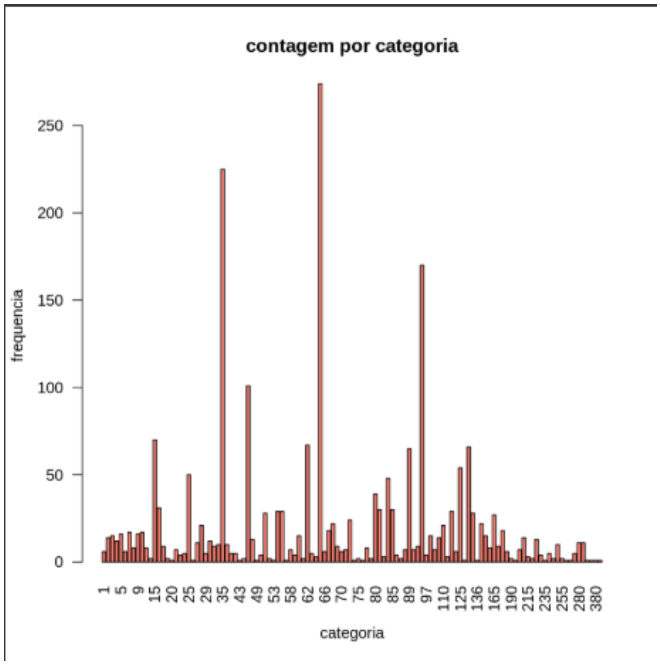
Figura 6– Gráfico de Pizza das CPUs



É realmente muito alto a quantidade de dados que estão entre 1 e 70, praticamente mais da metade dos valores desse gráfico.

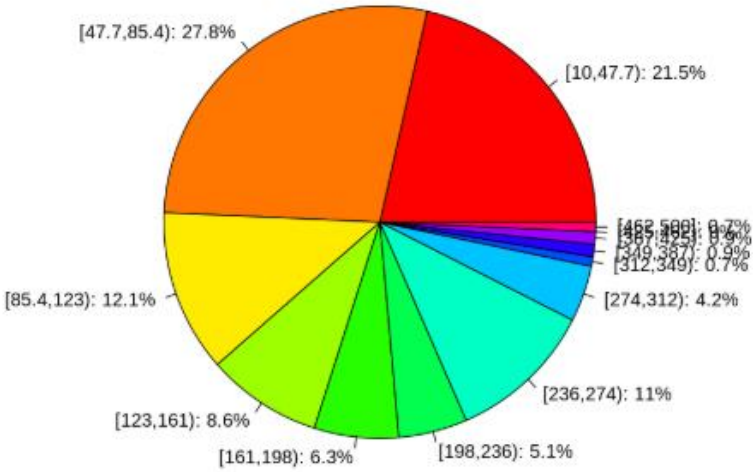
Logo abaixo também representa alguns valores onde é contado valores que são iguais na tabela.

Figura 7– Gráfico de Colunas das CPUs



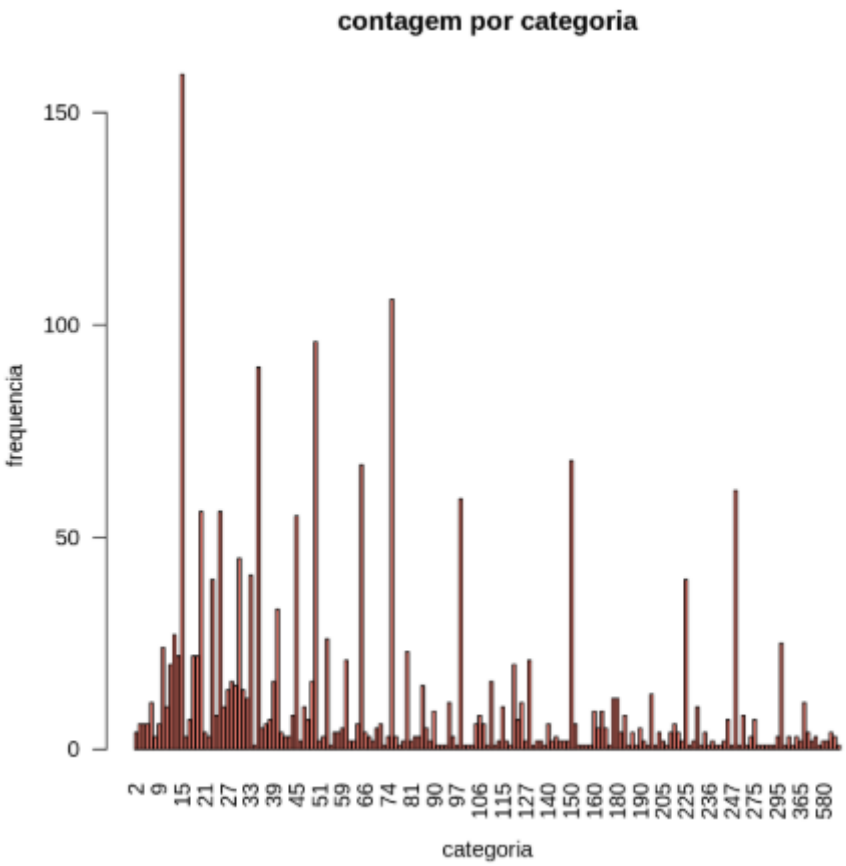
Há 3 grandes picos na tabela 65, 35, 95, que são números que provavelmente representam valores que são padrões na fabricação de processadores.

Figura 8– Gráfico de Pizza das GPUs



Aqui a faixa de valores que se encontra de 10 a 85.4 praticamente tem o valor da metade do *dataset* das GPUS.

Figura 9– Gráfico de Colunas das GPUs



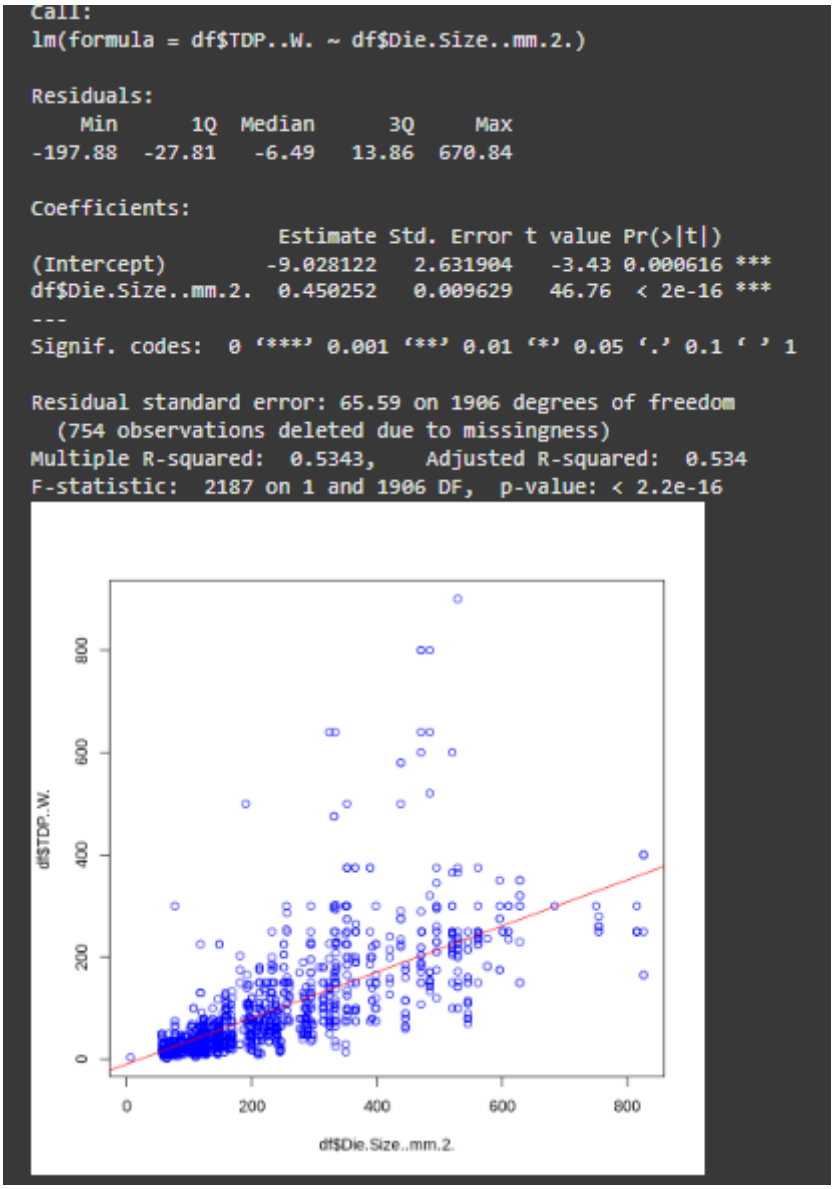
Assim como no gráfico anterior de barras, esse também possuí alguns picos, porém a variedade de picos é maior do que a quantidade anterior, porém as maiores frequências também são menores.

3.5 Regressões:

3.5.1 Regressão Linear simples:

A Regressão linear simples foi utilizada apenas duas variáveis para traçar uma reta média entre os valores:

Figura 10– Gráfico de Regressão Linear das GPUs



Como parâmetros foram utilizadas as variáveis TDP, que representa a quantidade de trabalho gasta em uma hora em energia elétrica (W), e o *Die.Size.mm.2* que representa a área do que essa GPU possui, ou seja, o tamanho que ela possui.

Com a visualização desses dados foi possível observar uma acurácia aproximada de 53.4%, onde é uma baixa precisão para utilizar como base, essa foi a

maior precisão dentro de todas as possíveis variáveis testadas na base de dados da GPU.

Agora passando para a análise no gráfico de regressão linear na base de dados da CPU, utilizando as mesmas variáveis os resultados foram bem diferentes. Assim como a regressão anterior foram utilizadas as melhores variáveis para essa comparação.

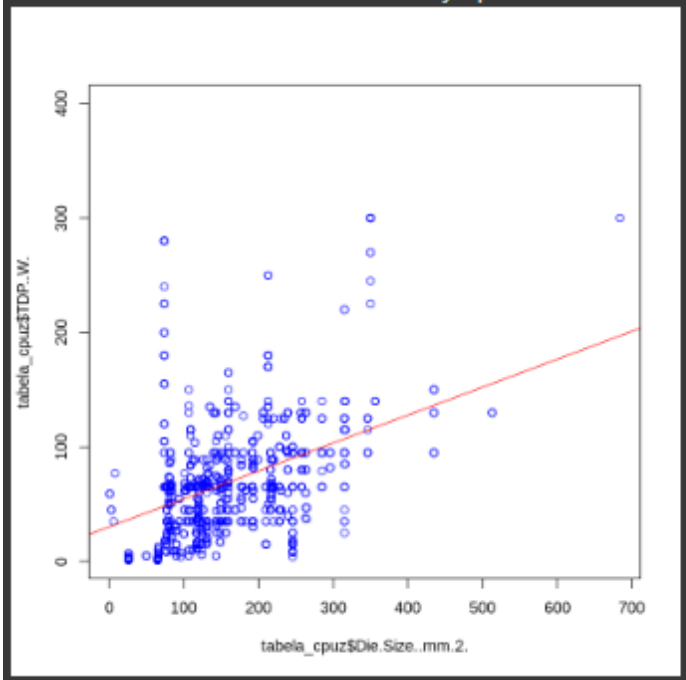
Figura 11– Gráfico de Regressão Linear das CPUs

```
Call:
lm(formula = tabela_cpuz$TDP..W. ~ tabela_cpuz$Die.Size..mm.2.)

Residuals:
    Min      1Q  Median      3Q     Max
-86.280 -27.221  -4.795  16.147 231.687

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 30.2575     2.4544   12.33   <2e-16 ***
tabela_cpuz$Die.Size..mm.2.  0.2440     0.0133   18.35   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43.18 on 1658 degrees of freedom
  (532 observations deleted due to missingness)
Multiple R-squared:  0.1688,    Adjusted R-squared:  0.1683
F-statistic: 336.8 on 1 and 1658 DF,  p-value: < 2.2e-16
```

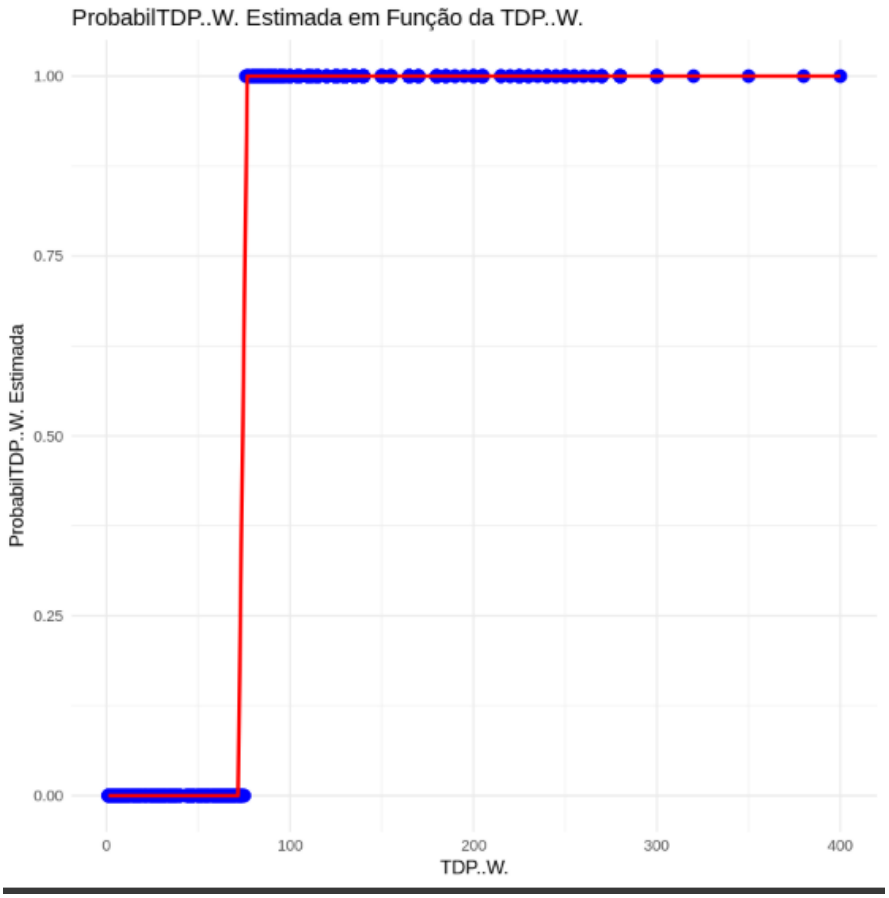


E o resultado foi pior que a regressão anterior, demonstrando apenas cerca de 16.83%, concluindo em ambas as regressões que para essa base de dados a regressão linear não terá tanta finalidade como os itens a seguir.

3.5.2 Regressão logística

Devido à grande falha nos testes de regressão logística, foi utilizado a regressão logística para fazer uma análise sobre esse TDP, conforme visualiza-se abaixo:

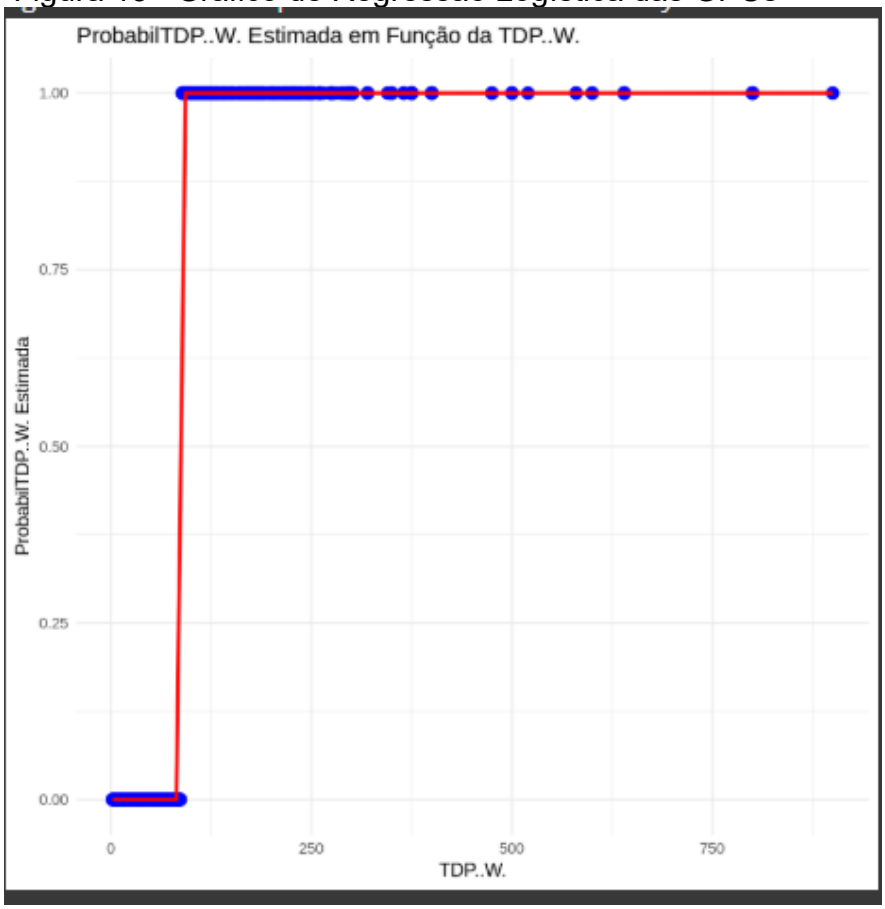
Figura 12– Gráfico de Regressão Logísticas das CPUs



Foi calculado a média de tdp dos processadores e utilizado como parâmetro para realizar essa regressão, conforme observado, é possível ver uma maior

quantidade encontrando-se abaixo dos 100 W, o que significa que há mais nessa faixa de processadores.

Figura 13– Gráfico de Regressão Logística das GPUs



Acima se observa essa regressão também para as GPUS, porém com essa mais amplitude nos dados, traz a média mais a trás, por isso o gráfico tem uma pequena diferença com o anterior.

4 ANÁLISE PREDITIVA E PRESCRITIVA

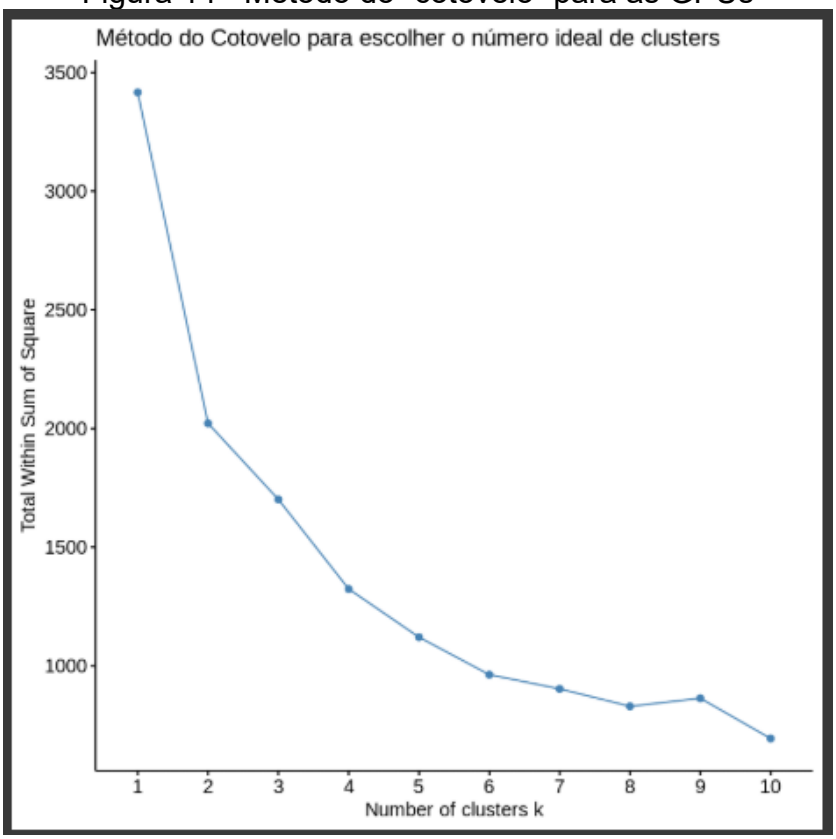
Para a análise preditiva foi utilizado *Machine Learning*, com alguns algoritmos de Aprendizado Supervisionado e Não supervisionado, sendo algoritmos como o PCA, *K-means*, *XGBOOST* e *Random Forest*, com os dois primeiros algoritmos foram

implementadas apenas para as GPUS, pois possuem todos os parâmetros na tabela, enquanto nos demais algoritmos foram implementadas para as duas tabelas.

4.2 *K-means*

Primeiro algoritmo implementado foi o *K-means* para identificar os grupos que tem mais importância dentro da base de dados.

Figura 14– Método do “cotovelo” para as GPUs



Primeiro foi realizado o teste do “Cotovelo”, que tem como finalidade descobrir a quantidade de *clusters* (grupos em inglês) que o método vai escolher para fazer as separações.

Forma testados 3 pontos, primeiro o ponto 2, que não foi muito efetivo, logo em seguida o k sendo como 8, onde o resultado foi muito bom, porém o melhor resultado foi quando foi usado 10 grupos.

Figura 15– Resultado do teste de *K-means*

```
K-means clustering with 10 clusters of sizes 17, 11, 34, 6, 80, 52, 10, 119, 51, 48

Cluster means:
   Process.Size..nm.       TDP..W. Die.Size..mm.2. Transistors..million.
1       -0.477905043  1.2181502208      2.49203443             1.8474724
2       -1.193788154  1.5892860327      1.46334192             2.1986473
3        2.826662533  0.0495665289     -0.02424485            -0.5632795
4       -1.356779398  2.4028384252      2.47738657             5.1478478
5        0.007819518  0.0034421985     -0.28707694            -0.3224086
6       -0.146186312 -1.0369422181     -0.57521856            -0.4056580
7       -0.400564099  1.7834186113      0.71153366             0.5447163
8       -0.052697259 -0.7316461200     -0.86247801            -0.6518892
9       -0.352909579  1.3439667251      0.96545577             0.6394233
10      -0.655388836  0.0008755418      0.55536419             0.3974216
    Freq..MHz. FP16.GFLOPS FP32.GFLOPS FP64.GFLOPS
1   0.31623430   1.2297661   1.3245690  3.47981245
2   0.83993874   1.6892652   3.9065181  0.07857898
3  -0.96884766  -0.4425397  -0.5199476 -0.28707988
4  -0.30133328   5.2489052   1.8848017  4.41983738
5   0.63637417  -0.3886085  -0.1556827 -0.27764646
6  -1.85996202  -0.4863993  -0.7871961 -0.36276617
7   0.43228862   0.8554984   0.7865942  2.30211761
8   0.14990131  -0.5159178  -0.6748578 -0.34611067
9   0.91968649   0.6931565   0.9995591 -0.11007439
10 -0.06507398   0.3736390   0.3277807 -0.24841964
```

Aqui é possível realizar uma análise mais concreta, logo acima na primeira linha, mostra respectivamente a quantidade de elementos que cada grupo possuí, e cada valor demonstrado na figura acima representa a centroide em relação à média, por exemplo, se o valor é 0 ele está na média, se ele é -1 está abaixo da média, se 1 está acima da média.

O primeiro grupo demonstra uma tecnologia um pouco melhor que a média, sendo possível observar pelo tamanho do chip (*Process.Size..mm.*) que representa o tamanho da tecnologia, quanto menor mais atual ele é, também é possível ver a quantidade de TDP, Transistores, FP16,FP32,FP64, com uma certa superioridade em relação à média geral.

Partindo para o segundo grupo visualiza-se uma tecnologia ainda mais recente que o anterior com um leve aumento no TDP, e uma diminuição do tamanho físico dele (*Die.Size..mm.2.*), e um aumento bem considerável na quantidade de transistores, também possuindo uma das frequências mais altas em relação as médias de todos os grupos.

Seguindo com o 3 grupo, ele provavelmente representa GPUS mais antigas devido a seus vários valores abaixo da média como transistores, Frequência, FP16, entre outros, além disso possuí um grande tamanho do chip, o que quer dizer que não é uma tecnologia tão recente.

O próximo grupo é o 4º grupo, que representa as maiores médias de FP16 e FP64, com um consumo muito mais alto que outras placas, provavelmente são as placas que são as melhores em questão de desempenho mesmo que possua uma tecnologia não tão recente.

Existem alguns grupos que estão mais próximos da média geral, grupos como 5, 8 e 10, onde valores se aproximam mais de 0.

O Grupo 6 representa placas de vídeo de provavelmente baixo custo, devido a sua média geral se aproximando mais de valores negativos do que positivos.

Seguindo com o grupo 7, visualiza-se valores levemente melhores do que a média, principalmente na parte dos *GFLOPS*, representando um grupo de boas GPUS intermediarias.

Por fim, o grupo 9 é um equilibro entre desempenho e construção física, pois possui valores levemente acima da média na maioria dos aspectos.

Figura 16 – Alguns valores do resultado do *K-means*

```
Within cluster sum of squares by cluster:
 [1]  40.16232  20.90803  86.76596  61.85573  74.46807  28.85974  29.49390
 [8]  62.88004 123.04729  84.09288
 (between_SS / total_SS =  82.1 %)
```

Analisando mais alguns aspectos, temos as somas dos quadrados de cada cluster, que representa valores que possuem mais ou menos variabilidade. Observando a figura é possível ver a alta variabilidade de alguns grupos como 5, 8,9 e 10, logo abaixo podemos ver a porcentagem de variabilidade total que foi explicada pelo algoritmo, sendo uma taxa de 82.1% que é considerada uma boa taxa.

4.3 PCA

Além do *K-means*, também foram realizados na alguns testes com PCA, que também é um algoritmo de agrupamento.

Figura 17 – Teste de PCA com *K-means*

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     2.1743 1.0606 0.9621 0.6963 0.50977 0.45644 0.39407
Proportion of Variance 0.5909 0.1406 0.1157 0.0606 0.03248 0.02604 0.01941
Cumulative Proportion  0.5909 0.7315 0.8472 0.9079 0.94033 0.96637 0.98578
                           PC8
Standard deviation     0.33724
Proportion of Variance 0.01422
Cumulative Proportion  1.00000
```

Aqui se observa o resultado dos testes, onde na primeira linha observa-se o desvio padrão do componente principal, e logo abaixo as duas últimas linhas representam a porcentagem de importância dessas variâncias de cada PC.

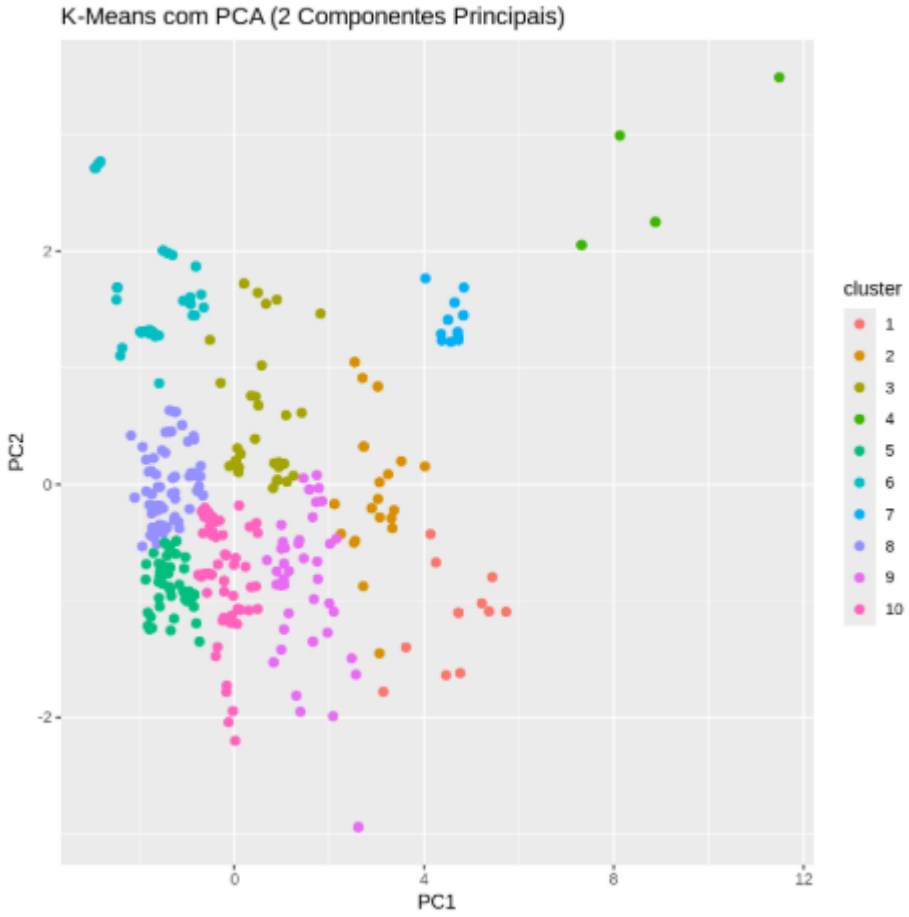
O PC1, PC2 e PC3, já explicam grande parte da tabela sendo cerca de 84%, onde com esses dados já seriam o suficiente para realizar uma boa análise reduzindo os ruídos com os demais.

4.4 PCA com *K-means*

Após a análise dos dois conjuntos de ferramentas de *Machine Learning* com aprendizado não supervisionado, foi realizado uma junção de ambos os algoritmos para a criação de uma análise mais visual conforme visualiza-se abaixo.

Para a análise foram utilizado o PC1 e PC2 que são os dois componentes principais mais importantes da tabela, com isso observa-se o comportamento de alguns grupos demonstrando valores que estão muito acima da média dos dois PCs, que é o caso dos grupos 4 e 7, o que assegura mais ainda as deduções vista no *k-means*, enquanto algumas placas que se encontram nos grupos 5,8 e 10, aproxima-se muito da média onde a análise se mantem ali.

Figura 18– Teste de PCA com *K-means*



4.5XGBOOST

Agora entrando no mundo dos algoritmos supervisionados, *XGBOOST* foi utilizado como um importante método de *Machine Learning* nessas tabelas.

Para os testes de CPU e GPU foram apresentadas duas respostas ao teste, que são os RMSE que mede o erro médio absoluto entre a previsão e o valor real, aonde os valores vão de 0 a 5 quanto maior o valor maior a taxa de erro, enquanto o R^2 mede a precisão que o algoritmo desempenhou.

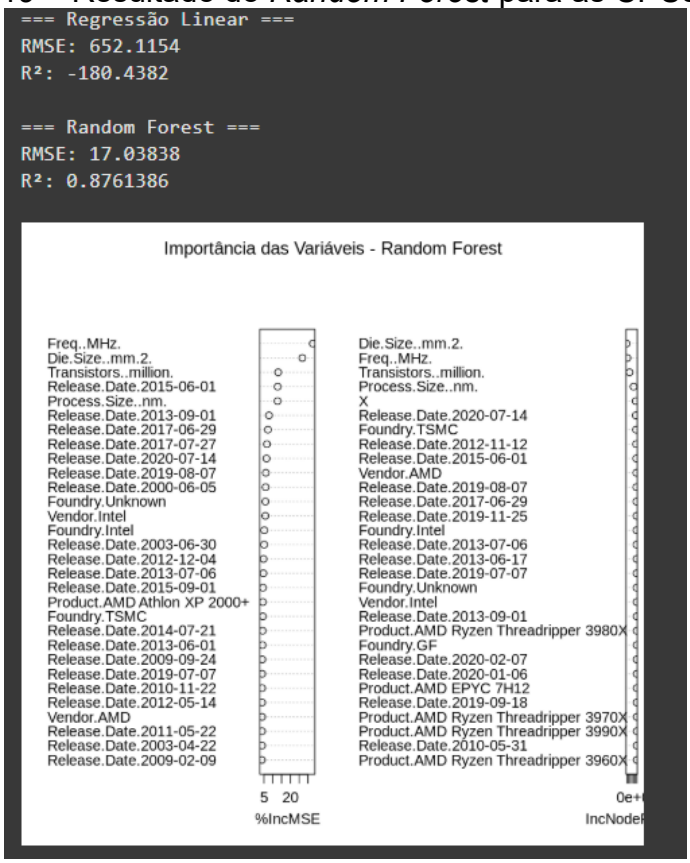
Para a CPU o algoritmo demonstrou um RMSE de 0,7356949, que indica um baixo erro na implementação, enquanto para R^2 demonstrou 0,9997691 que é uma precisão altíssima. Também para a GPU foram realizados os mesmos testes, RMSE demonstrou um valor de 1.355777 onde também indica uma baixa taxa de erro,

enquanto o R^2 demonstrou 0.999774 que é uma taxa muito alta e como na CPU chega muito próximo a perfeição.

4.6 Random Forest

Aqui temos um algoritmo que vai contribuir bastante para descobrir qual é a melhor GPU e CPU do *dataset*, o *Random Forest* irá ajudar a descobrir a importância das variáveis baseada em uma variável alvo.

Figura 19 – Resultado do *Random Forest* para as CPUs



Começando com a CPU primeiro foi realizado um teste para regressão linear e como visto nos gráficos anteriormente foi um teste com resultado horrível para essa base de dados, então é algo a ser descartado.

Logo em seguida temos o desempenho do *Random Forest* que traz um RMSE muito bom onde é possível visualizar alguns parâmetros nesse contexto para o RMSE:

Figura 20 – RMSE referencial no modelo do *Random Forest*

Faixa de RMSE	Significado no contexto técnico
RMSE < 10	Modelo extremamente preciso
RMSE entre 10–100	Modelo muito bom
RMSE entre 100–300	Modelo aceitável (bom ponto de partida)
RMSE entre 300–1000	Modelo com erros visíveis (pode melhorar)
RMSE > 1000	Modelo ruim (provavelmente sub ou overfitting)

Observando alguns valores é possível identificar um modelo muito bom para ser observado para esse teste, logo após isso pode-se ver o R^2 que mostra cerca de 87,6% de precisão demonstrando um alto percentual.

Com isso podemos observar o modelo em sí, esse modelo organiza as variáveis em posições do mais importante ao menos relevante de acordo com uma variável alvo, nesse caso a variável é TDP, ou seja, quando maior o TDP as variáveis que mais terão influência serão frequência e o tamanho físico do processador. Isso já será o suficiente para realizar um ranking para descobrir qual é o melhor processador baseado no TDP.

Para a construção desse *ranking* foi retirado através do *Random Forest* as variáveis de maior impacto no TDP, e com isso foi realizado uma média ponderada considerando um peso em porcentagem, para o próprio TDP como é uma variável mais importante foi definido o peso dela como 40%, e o restando dos 60% foram divididas nas demais variáveis, que foram consideradas como: quantidade de transistor em 30%, tamanho real do processador em 20% e 10% para o tamanho do chip, nessa tabela demonstrou alguns valores, sendo todas as variáveis daquela tupla na tabela e o Score_Ponderado, que é uma porcentagem onde ele atingiu sobre o peso dos valores, onde 1 seria o processador perfeito nesse mundo e 0 seria o pior processador possível a ser produzido.

Figura 21 – Ranking final para as CPUs

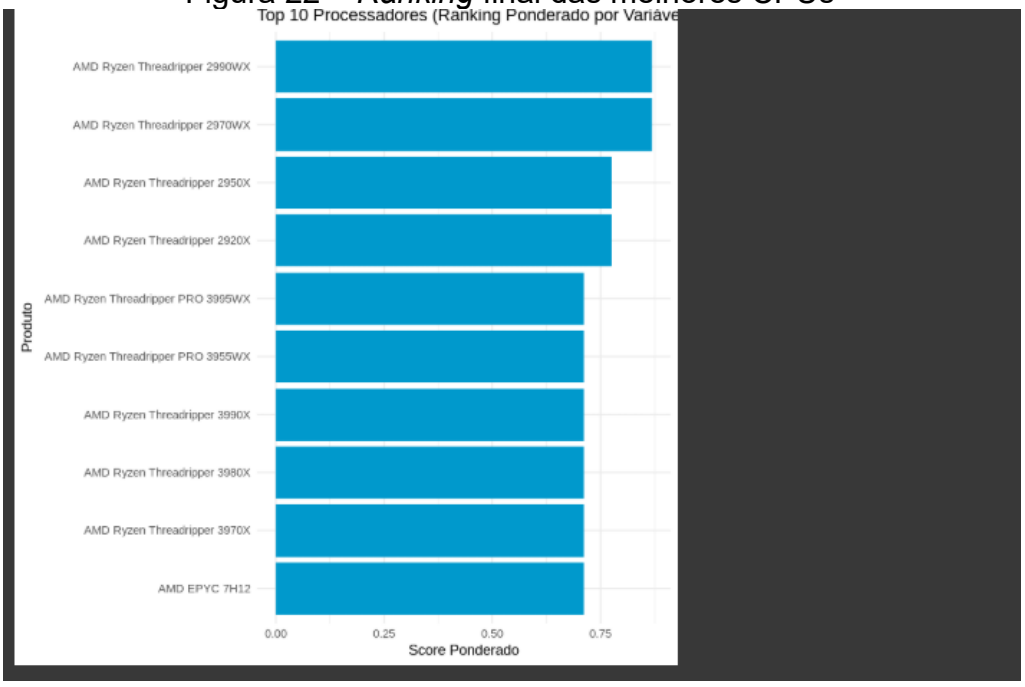
	Rank	Product	Score_Ponderado	TDP..W.
	<int>	<fctr>	<num>	<int>
1:	1	AMD Ryzen Threadripper 2970WX	0.8681411	250
2:	2	AMD Ryzen Threadripper 2990WX	0.8681411	250
3:	3	AMD Ryzen Threadripper 2920X	0.7744956	180
4:	4	AMD Ryzen Threadripper 2950X	0.7744956	180
5:	5	AMD Ryzen Threadripper PRO 3955WX	0.7107783	280

1539:	1539	AMD Opteron 840 EE	0.2125554	30
1540:	1540	Intel Pentium III 933	0.2117590	27
1541:	1541	Intel Pentium 4 1.4	0.2090687	55
1542:	1542	Intel Pentium 4 1.4	0.2090687	55
1543:	1543	Intel Pentium 4 1.3	0.2050553	52

	Transistors..million.	Die.Size..mm.2.	Process.Size..nm.	Freq..MHz.	Vendor
	<int>	<int>	<int>	<int>	<fctr>
1:	19200	213	12	3000	AMD
2:	19200	213	12	3000	AMD
3:	19200	213	12	3500	AMD
4:	19200	213	12	3500	AMD
5:	3800	74	7	3900	AMD

1539:	106	193	130	1400	AMD
1540:	44	80	180	933	Intel
1541:	42	217	180	1400	Intel
1542:	42	217	180	1400	Intel
1543:	42	217	180	1300	Intel

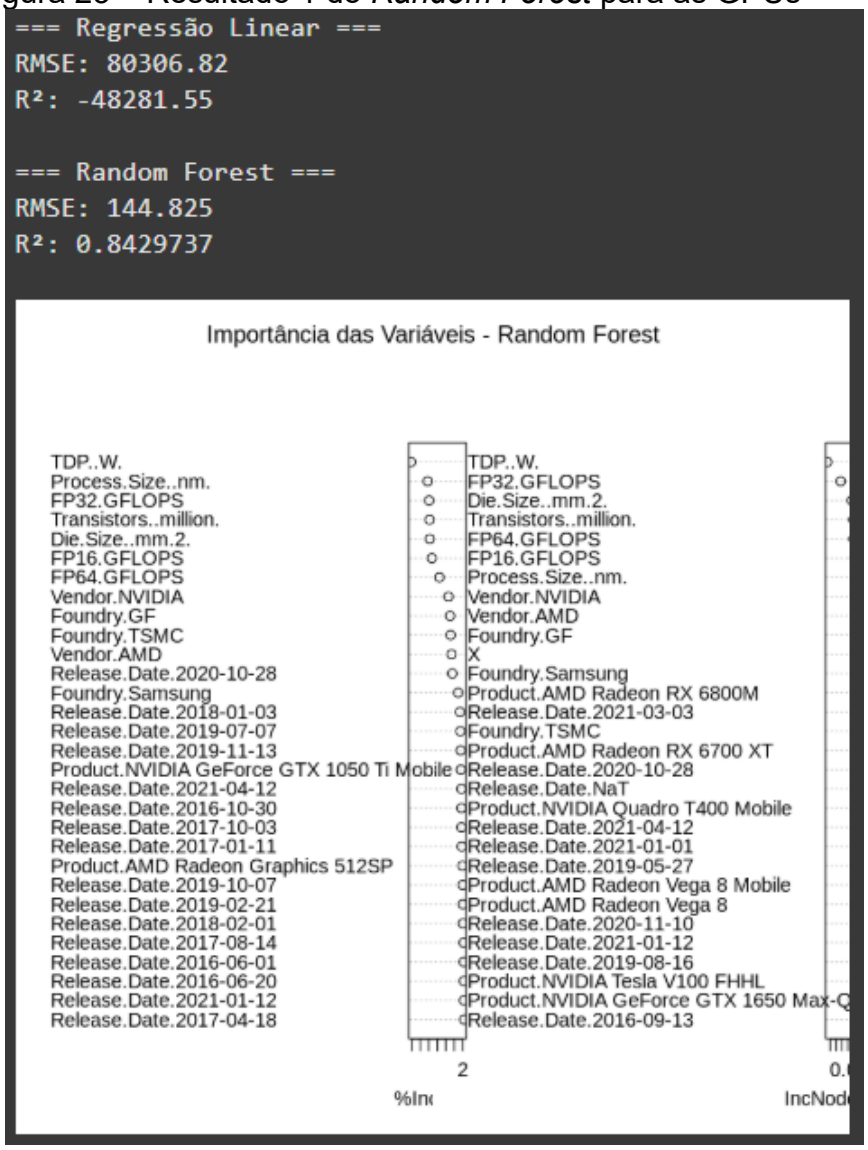
Figura 22 – *Ranking* final das melhores CPUs



E com isso gerou esse ranking, que demonstrou que *AMD Ryzen Threadripper 2970WX* é o melhor processador dessa lista, porém *AMD Ryzen Threadripper 2990WX* tem a mesma pontuação então ambos os processadores nesse caso podem ser considerados como os melhores.

Além da análise na CPU foram realizados os testes para GPU também onde foi realizado o teste do *Random Forest* duas vezes, considerando duas variáveis como alvo, sendo elas TDP e Freq.

Figura 23 – Resultado 1 do *Random Forest* para as GPUs



Nesse teste acima foi considerado a variável frequência, com um RMSE muito bom e o alto percentual do R^2, devido a grande quantidade de variáveis a média ponderada nesse caso vai funcionar diferente:

Figura 24 –Peso das variáveis no primeiro teste das GPUs

Variável	Peso Proposto
Freq..MHz.	0.3
FP32.GFLOPS	0.25
FP64.GFLOPS	0.2
Transistors..million.	0.1
Die.Size..mm.2.	0.10 (inverso)
Process.Size..nm.	0.05 (inverso)

Esse é o peso das variáveis, alguns valores foram definidos como negativos pois quando menor o tamanho melhor.

Figura 25 – *Ranking* 1 para as GPUs

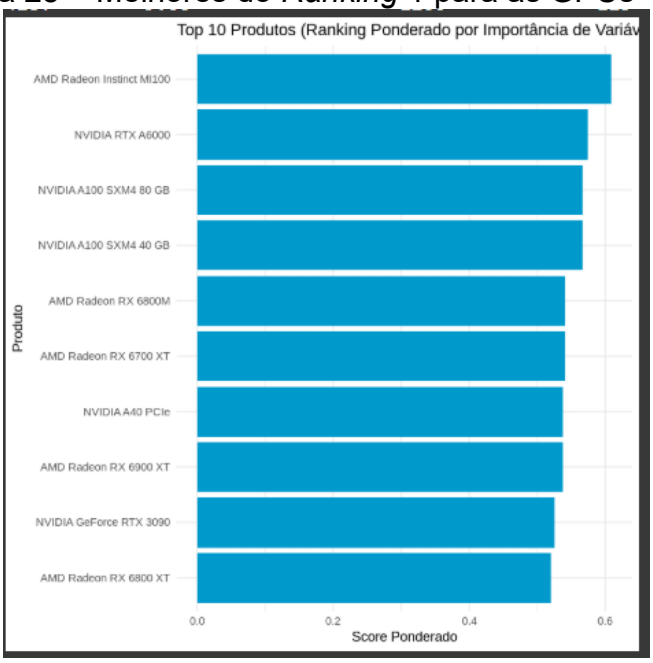
	Rank	Product	Score_Ponderado	Freq..MHz.	FP32.GFLOPS
	<int>	<fctr>	<num>	<int>	<num>
1:	1	AMD Radeon Instinct MI100	0.60911564	1000	23070.0
2:	2	NVIDIA RTX A6000	0.57414136	1455	40000.0
3:	3	NVIDIA A100 SXM4 80 GB	0.56679787	1095	19490.0
4:	4	NVIDIA A100 SXM4 40 GB	0.56679787	1095	19490.0
5:	5	AMD Radeon RX 6700 XT	0.54159335	2321	13210.0

424:	424	AMD Radeon R5 Mobile Graphics	0.10039127	200	553.0
425:	425	AMD Radeon R5 Mobile Graphics	0.09451951	200	325.2
426:	426	AMD Radeon R4 Mobile Graphics	0.09397716	200	251.5
427:	427	AMD Radeon R3 Mobile Graphics	0.09336052	200	167.7
428:	428	AMD Radeon R2 Mobile Graphics	0.09325679	200	153.6

	FP64.GFLOPS	Transistors..million.	Die.Size..mm.2.	Process.Size..nm.	Vendor
	<num>	<int>	<int>	<int>	<fctr>
1:	11540.00	50000	750	7	AMD
2:	1250.00	28300	628	8	NVIDIA
3:	9746.00	54200	826	7	NVIDIA
4:	9746.00	54200	826	7	NVIDIA
5:	825.90	17200	335	7	AMD

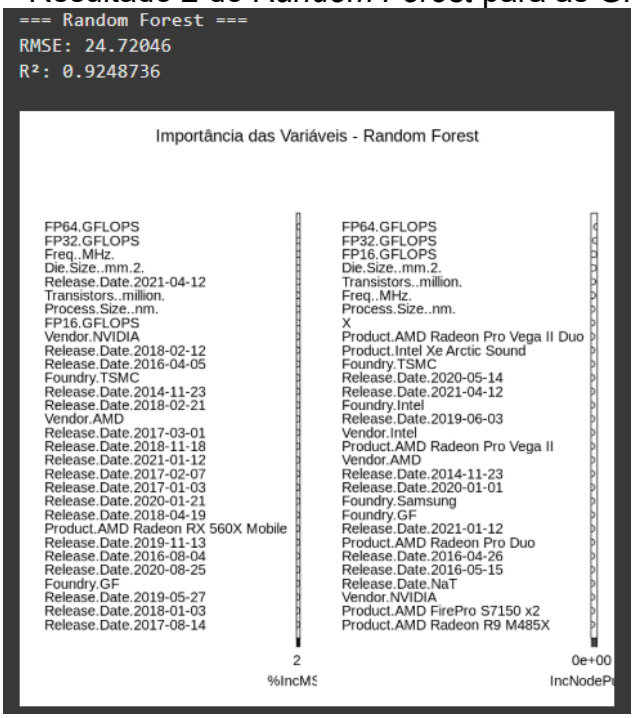
424:	276.50	1200	125	28	AMD
425:	20.33	1200	125	28	AMD
426:	15.72	1200	125	28	AMD
427:	10.48	1200	125	28	AMD
428:	9.60	1200	125	28	AMD

Figura 25 – Melhores do *Ranking* 1 para as GPUs



Esses foram os resultados, o percentual do maior score_ponderado, não foi tão grande quanto o valor das CPUs, devido a isso foi trocado a variável para TDP, para observar se o teste seria diferente:

Figura 26 – Resultado 2 do *Random Forest* para as GPUs



Com isso pode-se observar um resultado diferente, porém utilizando outros pesos já que a variável base não foi igual:

Figura 27 –Peso das variáveis no primeiro teste das GPUs

Variável	Peso Proposto
TDP..W.	0.2
FP32.GFLOPS	0.1
Freq..MHz.	0.05
Die.Size..mm.2.	0.10 (inverso)
Transistors..million.	0.1
Process.Size..nm.	0.10 (inverso)
FP64.GFLOPS	0.25

Então com esses pesos o *ranking* foi gerado, conforme é possível visualizar abaixo:

Figura 28 –*Ranking* 2 para as GPUs

	Rank	Product	Score_Ponderado	FP64.GFLOPS
	<int>	<fctr>	<num>	<num>
1:	1	NVIDIA A100 SXM4 80 GB	0.67728968	9746.00
2:	2	NVIDIA A100 SXM4 40 GB	0.67728968	9746.00
3:	3	AMD Radeon Instinct MI100	0.65610375	11540.00
4:	4	NVIDIA GRID A100A	0.61009830	6947.00
5:	5	NVIDIA GRID A100B	0.61009830	6947.00

424:	424	AMD Radeon R5 Mobile Graphics	0.10194897	276.50
425:	425	AMD Radeon R5 Mobile Graphics	0.09693475	20.33
426:	426	AMD Radeon R4 Mobile Graphics	0.09666984	15.72
427:	427	AMD Radeon R3 Mobile Graphics	0.09636866	10.48
428:	428	AMD Radeon R2 Mobile Graphics	0.09631802	9.60

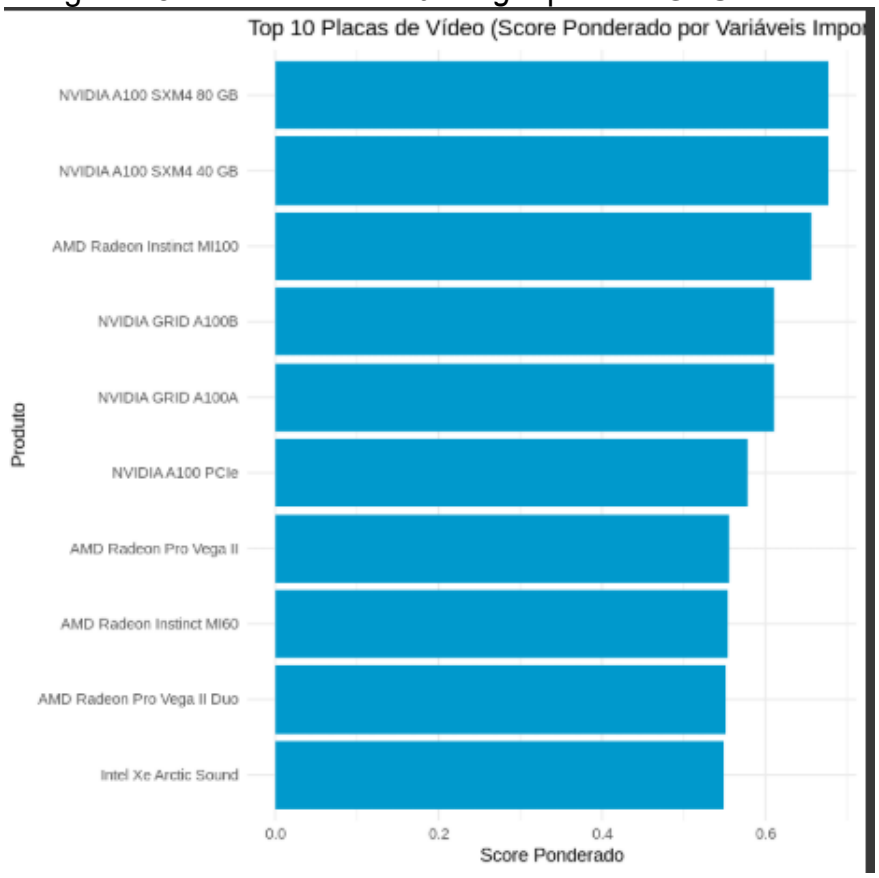
	FP32.GFLOPS	Freq..MHz.	Transistors..million.	TDP..W.	Die.Size..mm.2.
	<num>	<int>	<int>	<int>	<int>
1:	19490.0	1095	54200	400	826
2:	19490.0	1095	54200	400	826
3:	23070.0	1000	50000	300	750
4:	13890.0	900	54200	400	826
5:	13890.0	900	54200	400	826

424:	553.0	200	1200	15	125
425:	325.2	200	1200	15	125
426:	251.5	200	1200	15	125
427:	167.7	200	1200	15	125
428:	153.6	200	1200	15	125

	Process.Size..nm.	Vendor
	<int>	<fctr>
1:	7	NVIDIA
2:	7	NVIDIA
3:	7	AMD
4:	7	NVIDIA
5:	7	NVIDIA

424:	28	AMD
425:	28	AMD
426:	28	AMD
427:	28	AMD
428:	28	AMD

Figura 25 – Melhores do *Ranking* 1 para as GPUs



O Score ponderado do maior valor teve um aumento de 7% em relação ao teste anterior, porém os valores também foram trocados o que representa que a variável alvo pode fazer muita diferença.

5 CONSIDERAÇÕES FINAIS

Após feita a análise de vários gráficos e informações que eram apenas dados, foi possível transpor isso para um conteúdo de fato, objetivo principal do artigo era descobrir qual era o melhor CPU e qual era a melhor GPU, com a análise principal do algoritmo *Random Forest* e várias outras métricas, foi constatado que o melhor CPU com a variável alvo TDP, é o *AMD Ryzen Threadripper 2970WX* e *AMD Ryzen Threadripper 2990WX* pois ambos desempenharam com o mesmo resultado nos

testes, e para as GPUS com a mesma variável alvo, a GPU que desempenhou melhor foi a NVIDIA A100 SXM4.

6 REFERÊNCIAS

ALPAYDIN, Ethem. *Introduction to Machine Learning*. 4. ed. Cambridge: MIT Press, 2020.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. Cambridge: MIT Press, 2016. Disponível em: https://www.deeplearningbook.org/. Acesso em: 20 jun. 2025.

JAIN, Anil K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, v. 31, n. 8, p. 651–666, 2010. Disponível em: https://doi.org/10.1016/j.patrec.2009.09.011. Acesso em: 20 jun. 2025.

JAMES, Gareth et al. *An Introduction to Statistical Learning: with Applications in R*. 2. ed. New York: Springer, 2021. Disponível em: https://www.statlearning.com. Acesso em: 20 jun. 2025.

JOLLIFFE, Ian T. *Principal Component Analysis*. 2. ed. New York: Springer, 2002.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, v. 1, p. 281–297, 1967.

MARR, Bernard. *Big Data: Using SMART Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance*. Chichester: Wiley, 2016.

MONTGOMERY, Douglas C.; RUNGER, George C. *Applied Statistics and Probability for Engineers*. 7. ed. Hoboken: Wiley, 2020.

PROVOST, Foster; FAWCETT, Tom. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. Sebastopol: O'Reilly Media, 2013.

R CORE TEAM. *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing, 2024. Disponível em: https://www.R-project.org/. Acesso em: 20 jun. 2025.

SEBESTA, Robert W. *Conceitos de Linguagens de Programação*. 10. ed. São Paulo: Pearson, 2012.

SHMUELI, Galit; KOPPIUS, Otto R. Predictive Analytics in Information Systems Research. *MIS Quarterly*, v. 35, n. 3, p. 553–572, 2011. Disponível em: https://www.jstor.org/stable/23042796. Acesso em: 20 jun. 2025.

TANENBAUM, Andrew S.; BOS, Herbert. *Modern Operating Systems*. 4. ed. Upper Saddle River: Pearson, 2015.

TRIOLA, Mario F. *Introdução à Estatística*. 12. ed. Rio de Janeiro: LTC, 2019.

WICKHAM, Hadley. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol: O'Reilly Media, 2017. Disponível em: https://r4ds.had.co.nz. Acesso em: 20 jun. 2025.

WICKHAM, Hadley; GROLEMUND, Garrett. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol: O'Reilly Media, 2017. Disponível em: https://r4ds.had.co.nz. Acesso em: 20 jun. 2025.

[Recebido/Received: Dezembro 18 2024; Aceito/Accepted: Janeiro 29, 2025]