

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
SISTEMAS DE INFORMAÇÃO**

**MATHEUS JOSÉ DOS REIS - 12211BSI271
RAFAEL RANDER SILVA - 11511BCC049
VICTOR HUGO GONÇALVES DE OLIVEIRA - 11821BSI258**

PADRÕES DE PROJETO EM APRENDIZAGEM DE MÁQUINA

1. Introdução:

Este relatório apresenta implementações de alguns algoritmos de aprendizagem de máquina contendo soluções computacionais baseadas nos padrões de projeto: State, Template e Strategy. Divide-se em desenvolvimento, diagramas UML, implementação dos algoritmos e considerações finais.

2. Desenvolvimento:

2.1. Rede Neural Artificial:

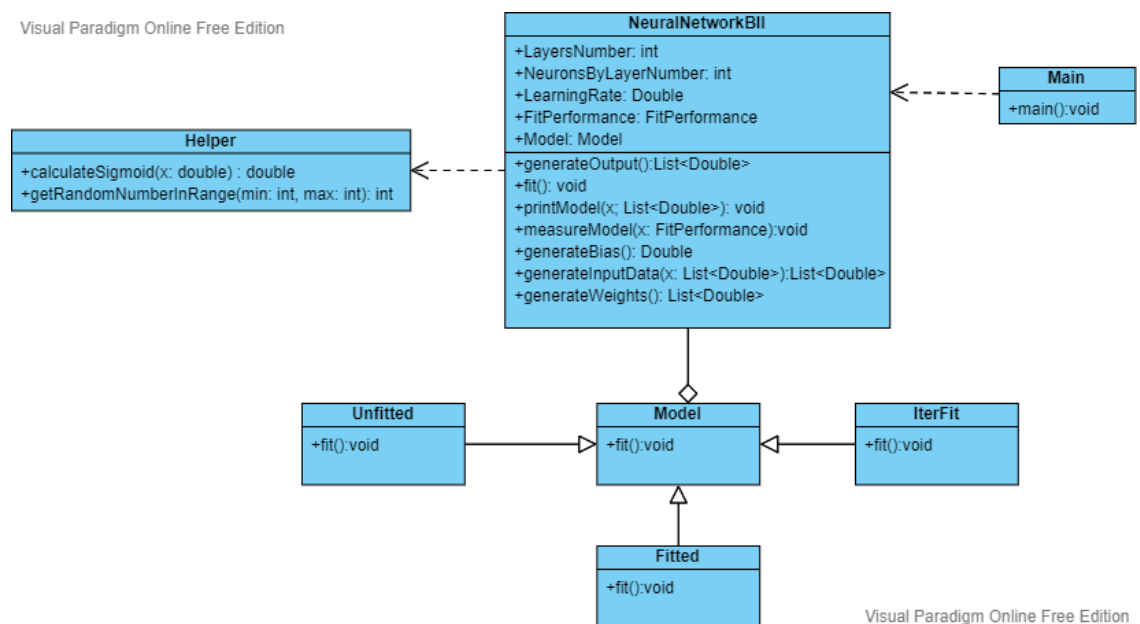
2.1.1 Padrão State:

Para o desenvolvimento desse algoritmo, foi-se utilizado o padrão de projeto *State* que modela um projeto dividindo um problema em questão por estados, no qual cada objeto é definido pelo seu estado interno, podendo ser trocado dependendo do seu comportamento em tempo de execução.

A maneira encontrada para abstrair o problema em questão nesse padrão de projeto foi definindo os possíveis modelos como sendo estados, sendo eles: modelo desajustado (*unfitted*), modelo ajustado (*fitted*) e modelo iterativo (*iterFit*).

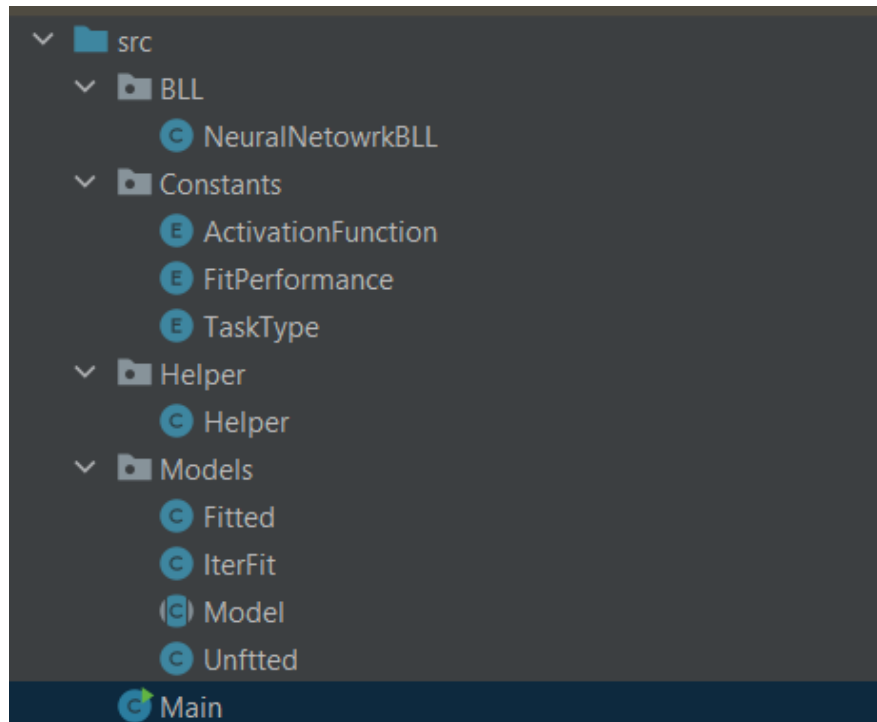
Sendo assim, a transição entre os estados se dá pela performance da execução do método *fit*.

2.1.2. Diagrama UML:



2.1.3. Implementação:

A estrutura pensada para a implementação desse projeto é como representada abaixo:



Tendo que o os estados estão estruturados no pacote de *Models*, os enumerators na pacote de *Constants* e no pacote BLL se encontra a *Context* de Rede Neural.

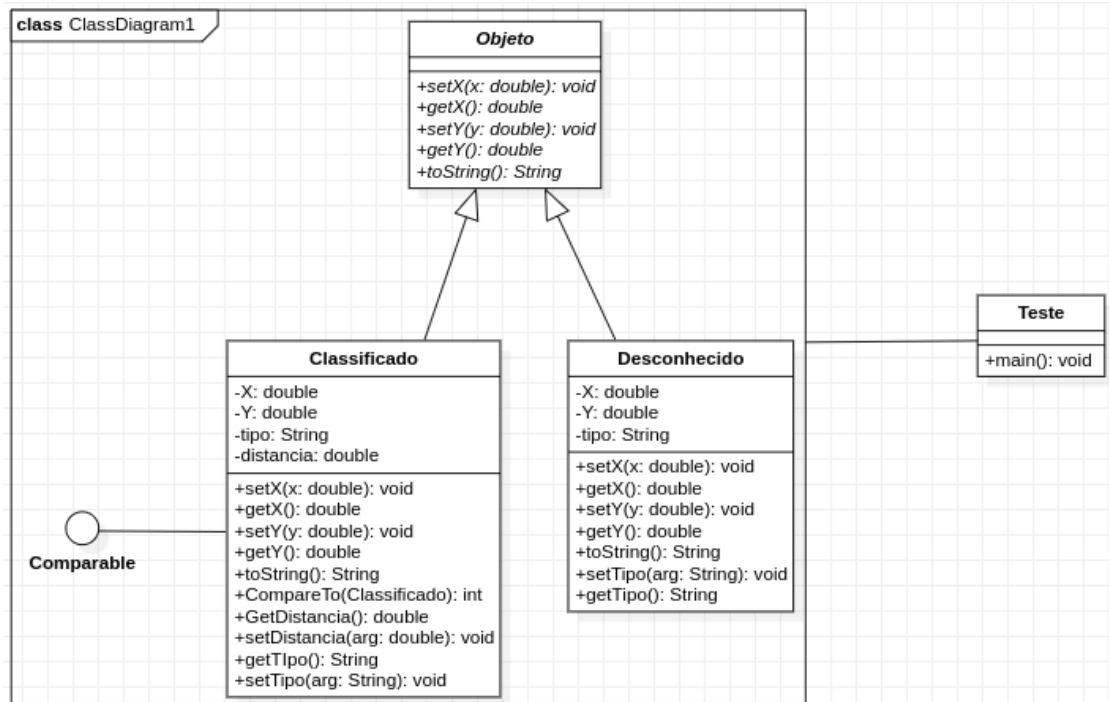
2.2. K-Vizinhos mais próximos (KNN):

2.2.1 Padrão Template:

Para a implementação deste algoritmo, foi utilizado o padrão Template, que basicamente definiu o esqueleto das operações utilizadas pelos métodos dos objetos do KNN. A classe abstrata *Objeto* definiu as operações modelos, e as subclasses concretas *Classificado* e *Desconhecido* às estenderam.

O Template Method é grandemente utilizado (principalmente em frameworks e APIs) por possibilitar termos um maior controle dos algoritmos e dos métodos que o mesmo utiliza. Também otimiza o código, evitando a sua reutilização.

2.2.2. Diagrama UML:



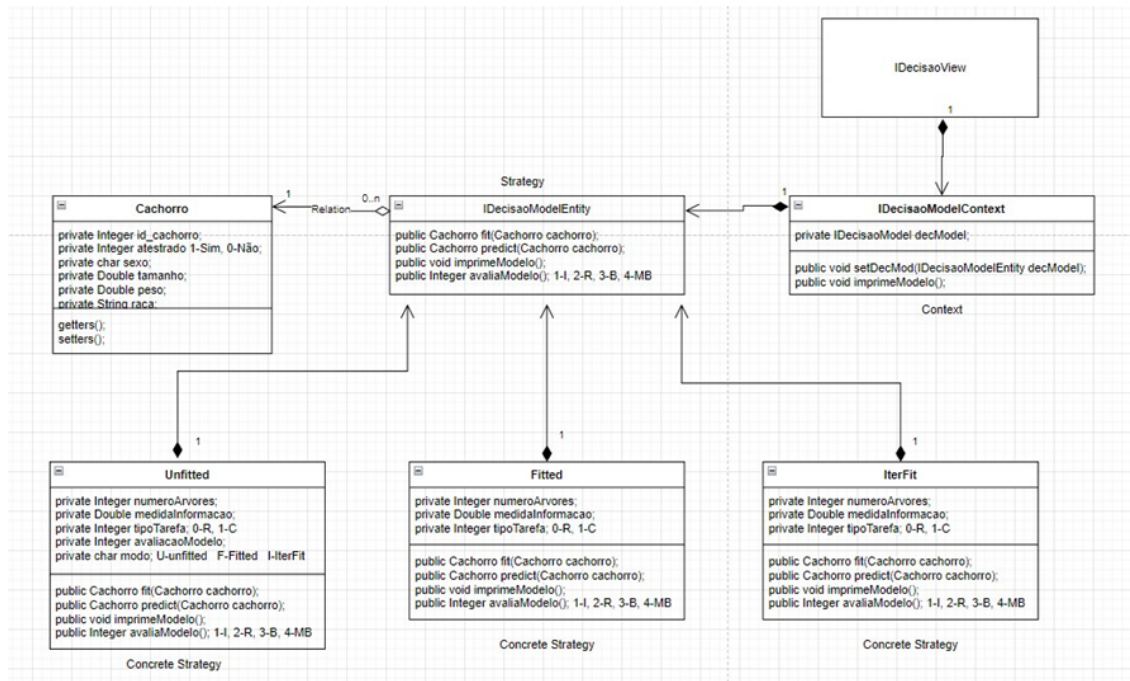
2.3. Árvore de decisão:

2.3.1. Padrão Strategy:

Para implementação desse algoritmo foi utilizado o padrão Strategy que é um padrão comportamental. Esse padrão delega as responsabilidades adquiridas pelas entidades, atribuindo, portanto, o comportamento.

O padrão Strategy tem como benefício a reutilização por parte do Contexto que permite escolher entre uma família de algoritmos que possuem funcionalidades em comum, com uso do Strategy há diminuição ou eliminação da lógica condicional clarificando ainda mais os algoritmos.

2.3.2. UML:



3. Considerações finais:

Em conclusão, durante a realização foi possível compreender a relevância de se conhecer os demais padrões de projetos devido à suas grandes vantagens, tendo que o projeto em que eles são bem utilizados ficam mais escaláveis e com maior compreensão para as próximas pessoas que tiverem acesso ao fonte, assim como o reuso de código que facilita no desenvolvimento. Ademais, foi possível se conhecer uma pequena, porém interessante parte da área de aprendizagem de máquina.