Gabriel Bodenmuller
Sérgio Venturi Pereira
Jorge Bandeo

Sistema Baseado em Casos para a previsão das notas

Itajaí - SC
2023

Gabriel Bodenmuller
Sérgio Venturi Pereira
Jorge Bandeo

Sistema Baseado em Casos para a previsão das notas

Relatório técnico sobre programação fuzzi apresentado como requisito parcial para composição de média 2 na disciplina de Inteligencia Artificial

Universidade do Vale do Itajaí - UNIVALI
Faculdade de Engenharia de Coputação
Bacharelado

Orientador: Prof. Rudimar Luíz Scaranto Dazzi

Itajaí - SC
2023

Sumário

1 Introdução

Neste trabalho, nossa equipe tem o objetivo de programar um protótipo de um Sistema Baseado em Casos (RBC - Case-Based Reasoning) para um tema definido por nós. Para isso, iremos determinar todos os componentes necessários para o funcionamento do RBC, desde a definição dos atributos até a recuperação dos casos.

As definições técnicas adotadas serão justificadas pela equipe, baseando-se em pesquisas na área. Para garantir a qualidade e efetividade do nosso protótipo, iremos embasar nossas decisões em estudos e referências relevantes sobre RBC.

O programa deve ser implementado em uma linguagem de programação de escolha da equipe, visto que os procedimentos são simples e podem ser implementados em qualquer linguagem. Será necessário cadastrar pelo menos 50 casos em uma base de dados. Além disso, o programa deve possuir uma interface que permita ao usuário alterar os pesos dos atributos, se desejado, e inserir um caso de entrada. A saída do programa deverá apresentar o caso de entrada, os pesos utilizados e uma lista com os casos da base ordenados por similaridade, do mais similar ao menos similar. Cada caso na lista deverá ser apresentado em uma linha, mostrando todos os atributos do caso e a porcentagem de similaridade com o caso de entrada.

A modelagem do RBC deverá indicar os atributos utilizados, os possíveis valores para cada atributo e como esses atributos serão comparados utilizando métricas de similaridade local.

1.1 Conceitualização do problema e sua resolução por RBC

Neste trabalho, nossa equipe propõe a utilização de um Sistema Baseado em Casos (RBC - Case-Based Reasoning) para realizar a previsão das notas dos períodos de um aluno. O tema selecionado consiste em utilizar um banco de dados que contém as notas dos períodos anteriores, bem como características individuais de cada aluno que podem influenciar em seus resultados acadêmicos. É importante ressaltar que este banco de dados está vinculado ao artigo (CORTEZ; SILVA, 2008), o qual utilizaremos como referência.

A abordagem do RBC se baseia no princípio de que situações similares têm resultados similares. Portanto, utilizaremos casos passados de alunos, juntamente com suas características individuais, para prever as notas de um aluno com base em casos semelhantes do passado.

Para a implementação do RBC, identificaremos os atributos relevantes que influenciam nas notas dos alunos. Isso pode incluir características como idade, gênero, escola de

origem, nível educacional dos pais, número de faltas, entre outros. Além disso, consideraremos as notas dos períodos anteriores como atributos importantes para a previsão.

Com base em pesquisas na área, definiremos métricas de similaridade adequadas para comparar os casos. Essas métricas podem levar em consideração a distância euclidiana entre os atributos numéricos, a similaridade de Jaccard para atributos categóricos, ou qualquer outra métrica apropriada para cada tipo de atributo.

A interface do programa permitirá que o usuário insira as características individuais do aluno atual e solicite a previsão de suas notas futuras. O programa utilizará o algoritmo de RBC para buscar os casos mais similares na base de dados, calcular uma média ponderada das notas desses casos e fornecer uma previsão das notas futuras do aluno.

Ao final do trabalho, entregaremos o programa implementado, juntamente com uma documentação detalhada da abordagem utilizada, justificando todas as definições técnicas adotadas. A apresentação do trabalho será realizada pelo grupo na data estabelecida, onde serão demonstrados os resultados obtidos e discutidos os desafios enfrentados durante o desenvolvimento do protótipo de RBC para a previsão de notas de alunos.

1.2 Escolha de Variáveis

Para realizar o tratamento dos dados do dataset selecionado, inicialmente foi necessário realizar uma filtragem das variáveis, uma vez que o dataset contém 33 características diferentes. Com o objetivo de ser fiel ao problema em questão, utilizamos informações fornecidas pelo criador (CORTEZ; SILVA, 2008) do dataset para definir nossas variáveis prioritárias e seus respectivos pesos. Para isso, baseamo-nos na seguinte descrição de importância relativa:

- C-Mat-Bin: failures (21.8%), absences (9.4%), schoolsup (7.0%), goout (6.5%), higher (6.4%)
- B-Por-Bin: G1 (22.8%), failures (14.4%), higher (11.9%), school (8.1%), Mjob (4.1%)
- C-Por-Bin: failures (16.8%), school (13.2%), higher (13.1%), traveltime (5.9%), famrel (5.7%)
- C-Mat-5L: failures (18.3%), schoolsup (9.5%), sex (5.7%), absences (5.6%), Medu (4.5%)
- C-Por-5L: failures (16.8%), higher (9.9%), school (9.3%), schoolsup (6.9%), Walc (6.6%)
- A-Mat-Reg: G2 (30.5%), absences (20.6%), G1 (15.4%), failures (6.7%), age (4.2%)
- B-Mat-Reg: G1 (42.2%), absences (18.6%), failures (8.9%), age (3.3%), schoolsup (3.2%)
- C-Mat-Reg: failures (19.7%), absences (18.9%), schoolsup (8.3%), higher (5.4%), Mjob (4.2%)
- C-Por-Reg: failures (20.7%), higher (11.4%), schoolsup (6.9%), school (6.7%), Medu (5.6%)

Como o conjunto de dados consiste em uma série de testes com variação nos pesos, decidimos realizar algumas operações para definir um único peso, sem perder a essência de sua significância. Para isso, optamos por separar cada tipo de teste e calcular a média dos valores dos pesos atribuídos. Dessa forma, obtemos uma medida da relevância geral das variáveis, levando em consideração as diferentes configurações de pesos utilizadas nos testes optendo o resultado abaixo.

- G2: 30.5%
- G1: 26.8%
- failures: 16.8%
- absences: 14.2%
- higher: 9.9%
- school: 9.3%
- schoolsup: 6.9%
- goout: 6.5%
- traveltime: 5.9%
- famrel: 5.7%
- sex: 5.7%
- Medu: 5.05%
- Mjob: 4.15%
- age: 3.75%

A soma das porcentagens fornecidas resulta em 136.25%. Para obter uma distribuição relativa das porcentagens, podemos normalizá-las dividindo cada valor pela soma total de 136.25%. Isso nos permitirá expressar as porcentagens como proporções relativas em relação ao total.

Realizando os cálculos, obtemos as seguintes proporções relativas para cada atributo:

- G2: 22.42
- G1: 19.64
- failures: 12.33
- absences: 10.43
- higher: 7.28
- school: 6.82
- schoolsup: 5.08
- goout: 4.77
- traveltime: 4.33
- famrel: 4.20
- sex: 4.20
- Medu: 3.71
- Mjob: 3.04
- age: 2.75

Para reduzir o número de variáveis a serem consideradas, selecionaremos as 11 mais relevantes com base nas porcentagens fornecidas. Em seguida, realizaremos a redistribuição da porcentagem para essas 11 variáveis selecionadas. Aqui está a nova distribuição de porcentagens após a seleção das variáveis mais relevantes:

- G2: 25.37
- G1: 22.24
- failures: 13.95
- absences: 11.78
- higher: 8.22
- school: 7.71
- schoolsup: 5.75
- goout: 5.39
- traveltime: 4.89
- famrel: 4.75
- sex: 4.75

2 Lógica de programação

Neste capítulo, descreveremos os métodos e implementações lógicas utilizadas no código para a realização da matriz de similaridade.

2.1 Funções de similaridade

Para as funções de similaridade, optamos por usar três tipos, sendo a numérica, a binária e a lógica. No programa, identificamos cada método da seguinte maneira:

```
"school": 2,
"sex": 2,
"failures": 3,
"absences": 1,
"higher": 2,
"schoolsup": 2,
"goout": 3,
"traveltime": 3,
"famrel": 3,
"G1": 3,
"G2": 3
```

1. Similaridade Numérica: Essas funções são aplicadas a atributos numéricos e medem a proximidade entre os valores. Alguns exemplos de funções de similaridade numérica incluem a distância Euclidiana, distância de Manhattan e coeficiente de correlação.

```
def SimilaridadeNumerica(F1,F2, max, min):
# 1 - | F1 - F2 | / (max - min)
a = F1 - F2 if F1 > F2  else F2 - F1 #| F1 - F2 |
return 1-a/(max - min)
```

2. Similaridade Binária: Essas funções são usadas para atributos binários, que têm apenas dois valores possíveis, como "sim"ou "não", "verdadeiro"ou "falso". As funções de similaridade binária comuns incluem a função de similaridade de Jaccard e o coeficiente de Sokal-Michener.

```
def SimilaridadeBooleana (F1,F2):
a = 1 if F1 == F2 else 0
return 1-(a)
```

3. Similaridade Lógica: Essas funções são aplicadas a atributos lógicos que possuem múltiplos valores possíveis. Elas comparam as diferentes combinações de valores lógicos atribuídos a cada atributo. Um exemplo comum de função de similaridade lógica é a função de similaridade de Tversky.

```
def SimilaridadeLolica(F1, F2, valores):
elementos= {}
distribuicao = 0
# distribui percentualmente os elemento no dicionario
for i in range(len(valores)):
    distribuicao += 1/len(valores)
    elementos[valores[i]] = distribuicao
# igaul ao numerico so que usa o auxilio do dicionarios

F1 = elementos[F1]
F2 = elementos[F2]
a = F1 - F2 if F1 > F2  else F2 - F1

return 1-a/(1)
```

2.2 Coleta e tratamento

Com as funções principais predefinidas, podemos realizar a coleta e filtragem de nosso banco de dados em formato CSV usando a biblioteca Pandas em Python. Aqui está um exemplo básico de como fazer isso:

```
    # L gica principal do programa
arquivo = pd.read_csv("Trabalho M2/Data set/student-mat.csv")
teste = pd.read_csv("Trabalho M2/Data set/datasetTestes.csv")
colunaG3 = arquivo["G3"]

# Selecionando colunas espec ficas nos DataFrames
arquivo = arquivo[["G2","G1", "failures","absences"
,"higher","school","schoolsup","goout","traveltime"
,"famrel","sex"]]
teste  = teste[["G2","G1", "failures","absences"
,"higher","school","schoolsup","goout","traveltime"
,"famrel","sex"]]

# Convers o das notas para o formato americano
for i in range(len(arquivo.values)):
    arquivo["G2"][i] = notaAmericana(arquivo["G2"][i])
    arquivo["G1"][i] = notaAmericana(arquivo["G1"][i])
    colunaG3[i] = notaAmericana(colunaG3[i])

for i in range(len(teste.values)):
    teste["G2"][i] = notaAmericana(teste["G2"][i])
    teste["G1"][i] = notaAmericana(teste["G1"][i])
```

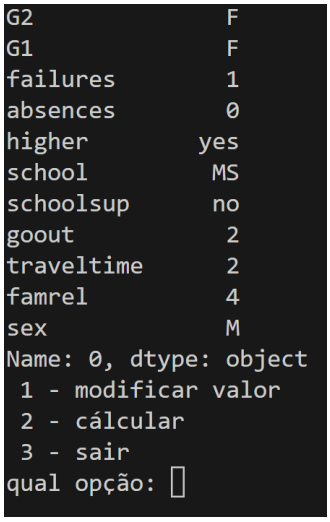
2.3 Calculador de similaridades

Após Coleta e tratamento, realizamos um menu onde permitimos ao usuário escolher entre calcular um exemplo pré-definido ou modificar as variáveis. Em seguida, é feito o cálculo da similaridade usando uma função dinâmica que implementa dois dicionários, que determinam o tipo de similaridade e seu respectivo peso, definidos anteriormente.

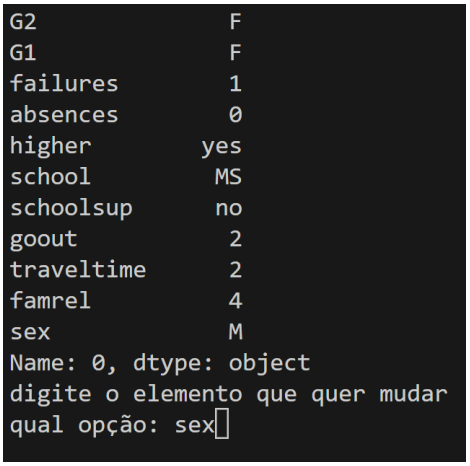
```
def CalculadorDeSimilaridade(Arquivo, predict):
# Obt m a lista das colunas do arquivo
colunas = Arquivo.columns.to_list()

# Lista para armazenar os valores de similaridade
semelhanca = []

# Calcula a soma dos pesos das colunas
pessom = 0
for i in colunas:
     pessom += pesos[i]

for i in range(0, len(Arquivo.values)):
    # Vari vel para armazenar a soma das similaridades
    somas = 0
    for j in colunas:
        # Verifica o tipo de similaridade da coluna
        if 1 == TS[j]:
            somas += pesos[j] * SimilaridadeNumerica(Arquivo[j][i],
            predict[j], maior(Arquivo, j), menor(Arquivo, j))
        elif 2 == TS[j]:
            somas += pesos[j] * SimilaridadeBooleana(Arquivo[j][i],
            predict[j])
        else:
            somas += pesos[j] * SimilaridadeLolica(Arquivo[j][i],
            predict[j], Arquivo[j].unique())

    # Adiciona o resultado da similaridade da linha    lista
    semelhanca.append(somas / pessom)

return semelhanca
```

3 Aplicação

A seguir, apresentamos a aplicação e seu funcionamento, juntamente com cada etapa do menu e a impressão dos primeiros elementos de maior similaridade.



Fonte: Autor



Fonte: Autor

```
G2              F
G1              F
failures        1
absences        0
higher        yes
school         MS
schoolsup      no
goout           2
traveltime      2
famrel          4
sex             M
Name: 0, dtype: object
os valores posiveis são:  ['F' 'M']
valor: F
```

Fonte: Autor

```
G2              F
G1              F
failures        1
absences        0
higher        yes
school         MS
schoolsup      no
goout           2
traveltime      2
famrel          4
sex             F
Name: 0, dtype: object
 1 - modificar valor
 2 - cálcular
 3 - sair
qual opção: 2
```

Fonte: Autor

```
G2              F
G1              F
failures        1
absences        0
higher        yes
school         MS
schoolsup      no
goout           2
traveltime      2
famrel          4
sex             F
Name: 0, dtype: object Resultado de maior similaridade:   F
    G2 G1  failures  absences higher school schoolsup  goout  traveltime  famrel sex G3        similaridade
250  F  F         1         0     no     GP        no      5           2       4   M  F  93.11324041811847%
78   F  F         3         2     no     GP       yes      1           2       4   M  D  92.71155632984905%
239  F  F         1         0     no     GP        no      4           1       5   M  F  91.22081881533101%
160  F  F         2         0     no     GP        no      2           2       3   M  F  90.29834494773519%
252  F  F         1         4     no     GP        no      5           1       3   M  F  89.84602206736353%
PQT
```

Fonte: Autor

4 Conclução

O trabalho proposto pela equipe consiste na programação de um protótipo de um Sistema Baseado em Casos (RBC) para a previsão das notas de períodos de alunos. A abordagem do RBC se baseia na premissa de que situações similares têm resultados similares, utilizando casos passados de alunos, juntamente com suas características individuais, para prever as notas de um aluno com base em casos semelhantes do passado.

A equipe definiu os atributos relevantes que influenciam nas notas dos alunos, considerando características como idade, gênero, escola de origem, nível educacional dos pais, número de faltas, entre outros. Além disso, foram definidas métricas de similaridade adequadas para comparar os casos, levando em consideração diferentes tipos de atributos, como numéricos, binários e lógicos.

O programa a ser implementado terá uma interface que permitirá ao usuário inserir as características individuais do aluno atual e solicitar a previsão de suas notas futuras. O algoritmo de RBC buscará os casos mais similares na base de dados, calculará uma média ponderada das notas desses casos e fornecerá uma previsão das notas futuras do aluno.

A escolha das variáveis relevantes foi baseada em uma análise das porcentagens atribuídas a cada atributo, considerando sua importância relativa. As variáveis selecionadas foram redistribuídas com base nessas porcentagens para determinar sua relevância geral no contexto do problema.

No capítulo sobre a lógica de programação, foram descritas as funções de similaridade utilizadas no código, incluindo funções numéricas, binárias e lógicas, adequadas para comparar os diferentes tipos de atributos presentes no sistema.

Ao final do trabalho, a equipe entregará o programa implementado, juntamente com uma documentação detalhada da abordagem utilizada, justificando todas as definições técnicas adotadas. Durante a apresentação do trabalho, serão demonstrados os resultados obtidos e discutidos os desafios enfrentados durante o desenvolvimento do protótipo de RBC para a previsão de notas de alunos.

Portanto, a conclusão é que o trabalho proposto busca implementar um sistema baseado em casos para prever as notas de períodos de alunos, utilizando atributos relevantes e métricas de similaridade apropriadas. A equipe pretende fornecer uma solução efetiva e demonstrar a aplicabilidade do RBC nesse contexto específico.

Referências

CORTEZ, P.; SILVA, A. M. G. Using data mining to predict secondary school student performance. EUROSIS-ETI, 2008. Citado 2 vezes nas páginas 3 e 5.