

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ARTHUR GUEDES DE SOUZA (213281)
GABRIEL DOMINGUES FERREIRA (216207)
MATHEUS CUMPIAN (222182)

PROUNI
Mineração de Dados

Limeira - SP
2019

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ARTHUR GUEDES DE SOUZA
GABRIEL DOMINGUES FERREIRA
MATHEUS CUMPIAN

PROUNI
Mineração de Dados

Trabalho apresentado à disciplina de
Mineração de Dados, da Faculdade de
Tecnologia da Universidade Estadual
de Campinas, sob orientação da Prof.
Ana Estela Antunes da Silva, Dr^a.

Limeira - SP
2019

SUMÁRIO

1. INTRODUÇÃO.....	4
2. O QUE É MINERAÇÃO DE DADOS?.....	4
2.1 AGRUPAMENTO.....	5
2.1.1 MEDIDAS DE SIMILARIDADE.....	5
2.1.1.1 MEDIDAS DE AVALIAÇÃO DE GRUPOS.....	5
2.1.1.1.1 COEFICIENTE DE SILHUETA.....	5
2.1.2 MÉTODOS DE AGRUPAMENTO.....	6
2.1.3 REPRESENTAÇÃO DO AGRUPAMENTO.....	6
2.1.4 MATRIZ DE CONTINGÊNCIA.....	7
2.1.4.1 ATRIBUTOS BINÁRIOS SIMÉTRICOS.....	8
2.1.4.2 ATRIBUTOS BINÁRIOS ASSIMÉTRICOS.....	8
2.1.5 SIMILARIDADE DE JACCARD.....	8
2.2 CLASSIFICAÇÃO.....	9
2.2.1 MODELO PREDITIVO.....	9
2.2.2 ÁRVORES DE DECISÃO.....	11
2.2.3 AVALIAÇÃO DE DESEMPENHO.....	11
2.2.4 CLASSIFICAÇÃO BINÁRIA.....	11
2.3 ASSOCIAÇÃO.....	12
2.3.1 REGRAS DE ASSOCIAÇÃO.....	13
2.3.2 SUPORTE E CONFIANÇA.....	13
3. DESCRIÇÃO DO PROBLEMA.....	13
3.1 EXPLICAÇÃO DOS ATRIBUTOS.....	14
3.2 EXPLICAÇÃO DO PRÉ-PROCESSAMENTO.....	16
3.2.1 PRÉ-PROCESSAMENTO PARA TAREFA DE ASSOCIAÇÃO.....	16
3.2.2 PRÉ-PROCESSAMENTO PARA TAREFA DE AGRUPAMENTO.....	16
3.3 EXPLICAÇÃO DA ESCOLHA DA TAREFA PARA O PROBLEMA.....	17
3.3.1 AGRUPAMENTO.....	17
3.3.2 CLASSIFICAÇÃO.....	17
3.3.3 ASSOCIAÇÃO.....	17
4. METODOLOGIA.....	18
4.1 PRÉ-PROCESSAMENTO.....	18
4.1.1 PREPARAÇÃO PARA AGRUPAMENTO.....	18
4.2 TAREFA DE MINERAÇÃO.....	22
4.2.1 AGRUPAMENTO.....	22
4.2.2 CLASSIFICAÇÃO.....	28
4.2.3 ASSOCIAÇÃO.....	29
4.2.3.1 PREPARAÇÃO PARA ASSOCIAÇÃO.....	29
4.2.3.2 COMO FOI FEITO?.....	29
4.2.3.3 PROBLEMAS E SOLUÇÕES.....	31

5. RESULTADOS E DISCUSSÃO	32
5.1 AGRUPAMENTO	32
5.2 CLASSIFICAÇÃO	34
5.3 ASSOCIAÇÃO	38
6. REFERÊNCIAS BIBLIOGRÁFICAS	39

1. INTRODUÇÃO

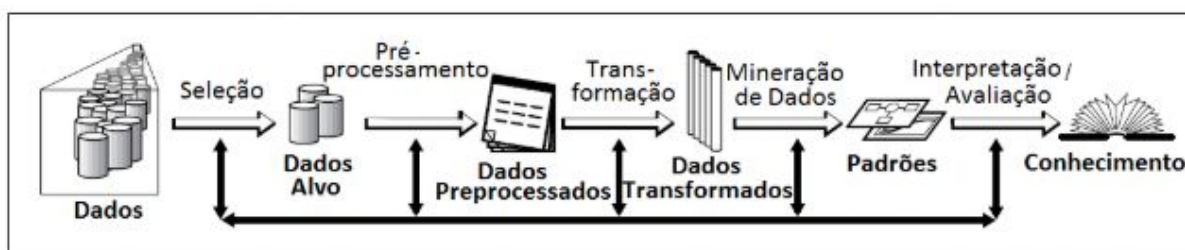
Com a grande coleta e armazenamento de dados, as organizações começaram a acumular quantidades brutas de informações. Porém, o processo de extrair conhecimento das grandes bases de dados, tem provado ser extremamente desafiador. Técnicas tradicionais de análise de dados não podem ser usadas devido ao tamanho do conjunto de dados ser muito grande.

[...] Diversas organizações, por exemplo, a Wal-Mart, TAM linhas aéreas, IBGE e a NASA, detém em seu departamento de tecnologia bases de dados de centenas de terabytes de informações. Sabendo-se que o armazenamento do maior volume possível de informações é benéfico para os seus processos, é aceitável deduzir que a dificuldade de interpretar e analisar esses dados são diretamente proporcionais à quantidade dos mesmos. (BRITO, 2012).

Muitas pesquisas estão sendo direcionadas para o desenvolvimento de técnicas com o objetivo de extrair o máximo de informações a partir de um grande volume de dados que foram adquiridos e armazenados durante o tempo, e transformar estas informações em conhecimento útil. Esta área é conhecida como KDD [1], sigla em inglês para Knowledge Discovery in Databases, ou Descoberta de Conhecimento em Bases de Dados.

O KDD é formado por etapas que, uma vez executadas, resultará na geração do conhecimento. Este processo é composto, conforme apresentado na Figura 1, por: seleção dos dados utilizados; o pré-processamento; transformação para um formato adequado; a mineração de dados e a análise dos resultados obtidos para a sua aplicação no processo decisório (interpretação/avaliação) [2].

Figura 1 - As fases do KDD



Fonte: DevMedia¹.

2. O QUE É MINERAÇÃO DE DADOS?

Mineração de dados (em inglês, data mining) é o processo de explorar bases de dados usando algoritmos adequados para obter conhecimento. Existem três tipos de tarefas de mineração: agrupamento, classificação e associação, e cada tarefa será explicada nos tópicos abaixo.

¹ Disponível em: <<https://www.devmedia.com.br/mineracao-de-dados-tarefas-e-tecnicas/30919>>. Acesso em: 03 jun. 2019.

2.1. AGRUPAMENTO

Um agrupamento (ou *cluster*) é uma coleção de registros próximos entre si e distantes dos outros registros de outros agrupamentos.

[...] agrupar objetos é o processo de particionar um conjunto de dados em subconjuntos (grupos) de forma que os objetos em cada grupo (idealmente) compartilhem características comuns [...] (DE CASTRO; FERRARI, 2016, p. 88).

Pode-se concluir que os objetos que são do mesmo grupo são mais similares entre si do que em relação a objetos que pertencem a outros grupos [3].

Em geral, a proximidade dos registros é calculada a partir de alguma medida de similaridade ou distância.

2.1.1. MEDIDAS DE SIMILARIDADE

O objetivo dos métodos de agrupamento é a formação de grupos em que a distância intragrupo seja menor que a distância intergrupo, a distância intragrupo é a similaridade (proximidade) e a intergrupo é dissimilaridade (distância), esses valores serão utilizados durante o agrupamento.

Uma estrutura de dados muito utilizada é a matriz de dissimilaridade, em que cada elemento corresponde a uma medida de distância entre pares de objetos. Essas medidas de dissimilaridade são muito utilizadas para avaliar a proximidade entre os objetos.

2.1.1.1. MEDIDAS DE AVALIAÇÃO DE GRUPOS

A avaliação da qualidade dos grupos formados é feita, normalmente, utilizando a similaridade inter e intragrupo, que são as medidas de coesão e separação:

- **Coesão:** A medida de coesão ou similaridade intragrupos mede quanto os objetos dentro de um grupo são semelhantes. Esse cálculo é feito a partir da similaridade de cada objeto ao centro do grupo.
- **Separação:** A medida de separação ou similaridade intergrupos mede quanto os grupos estão separados entre si, por exemplo, a distância dos centros de um par de grupos.

2.1.1.1.1. COEFICIENTE DE SILHUETA

O coeficiente de silhueta combina as medidas de coesão e separação, ela verifica o quão bem os objetos estão incluídos no seu grupo. O coeficiente é calculado a partir da fórmula:

$$s(x_i) = \frac{(b(x_i) - a(x_i))}{\max\{a(x_i), b(x_i)\}}$$

$a(x_i)$: distância média entre x_i e todos os objetos de seu grupo (medida de coesão).

$b(x_i)$: distância média entre x_i e outro grupo ao qual x_i não pertence (medida de separação).

Com os valores do coeficiente de silhueta de todos os objetos, é possível calcular o valor de silhueta global p (S_p), a partir da fórmula [4]:

$$S_p = \sum_{i=1}^n \frac{s(x_i)}{n}$$

Sendo $s(x_i)$ o coeficiente de silhueta do objeto i e n o número de objetos do grupo.

2.1.2. MÉTODOS DE AGRUPAMENTO

Existem diversos tipos de métodos de agrupamento, mas de forma abrangente, esses métodos podem ser divididos em hierárquicos e particionais.

Nos métodos hierárquicos são produzidos sequências de partições aninhadas, e eles podem ser aglomerativos ou divisivos. Os métodos aglomerativos começam com um objeto que pertencendo a um grupo, os pares de objetos mais próximos são unidos até se formar um grupo que contenha todos os objetos. Já os métodos divisivos inicia-se com um grupo contendo todos os objetos, e então esse grupo é dividido até que sejam gerados grupos com apenas um objeto.

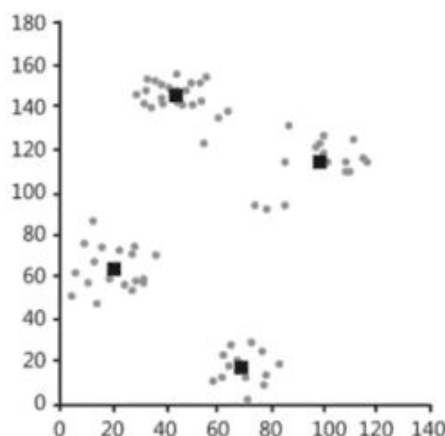
Nos métodos particionais, é dado um conjunto com n objetos, e nesse método é construído k partições de dados, sendo que cada uma representa um cluster, em que $k \leq n$. A partir disso, é criada uma partição inicial e os objetos são realocados iterativamente entre os grupos com o objetivo de melhorar o particionamento.

2.1.3. REPRESENTAÇÃO DO AGRUPAMENTO

Depois que os grupos forem gerados, eles podem ser representados de três maneiras diferentes. Essas formas são os protótipos, grafos e árvores [4]:

- **Protótipos:** São a principal maneira de representação dos grupos, que são através dos centróides (pontos médios).

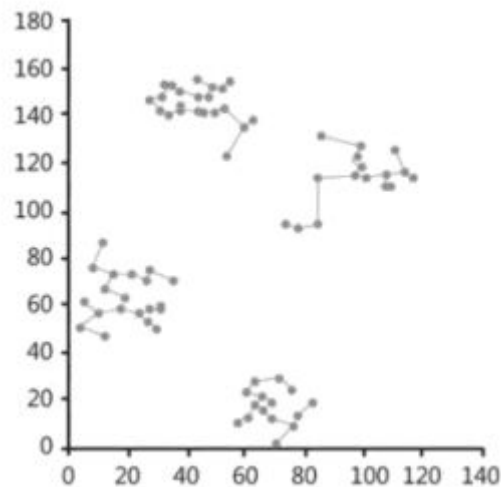
Figura 2 - Representação de protótipos



Fonte: DE CASTRO, 2016, p.107

- **Estruturas em grafos:** O grafo é um conjunto de nós e arcos, cada nó é um objeto e os arcos são as conexões entre os objetos. Os objetos conectados formam um grupo, esse grupo corresponde a um subgrafo, o conjunto de todos os subgrafos forma o grafo como um todo.

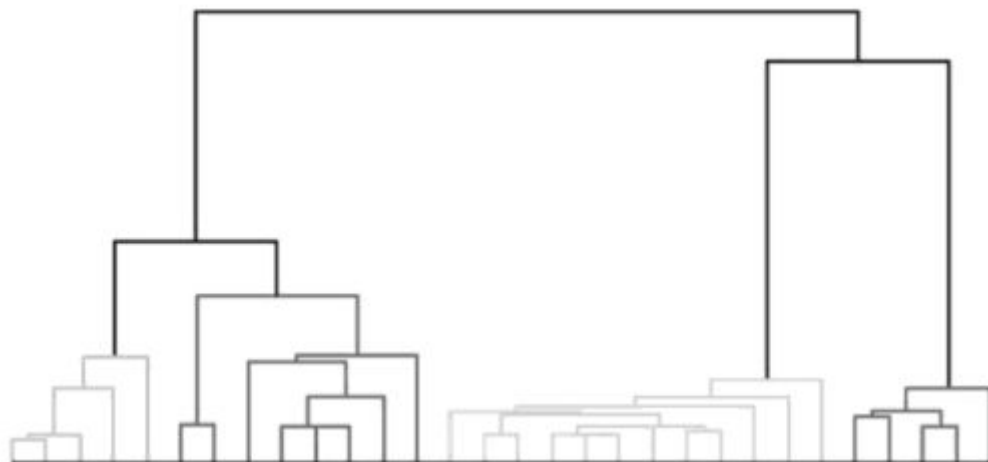
Figura 3 - Representação de grafos



Fonte: DE CASTRO, 2016, p.107

- **Estruturas em árvore:** Essa estrutura fornece o tipo de representação hierárquica das relações entre os objetos e grupos.

Figura 4 - Representação de árvores



Fonte: DE CASTRO, 2016, p.107

2.1.4. MATRIZ DE CONTINGÊNCIA

A matriz de dissimilaridade é determinada a partir de valores binários (0 ou 1, sim ou não), e esses valores são tratados de forma a calcular a distância entre eles. Essa distância é calculada a partir da matriz de contingência.

		Objeto j		
		1	0	Soma
Objeto i	1	q	r	q + r
	0	s	t	s + t
	Soma	q + s	r + t	p

q : número de atributos com valor 1 para i e j ;

r : número de atributos com valor 1 para i e 0 para j ;

s : número de atributos com valor 0 para i e 1 para j ;

t : número de atributos com valor 0 para i e 0 para j ;

p : número total de atributos;

2.1.4.1. ATRIBUTOS BINÁRIOS SIMÉTRICOS

Um atributo booleano é simétrico se o valor 0 ou 1 possuir a mesma importância para a análise de grupos. A distância entre atributos binários simétricos é dada por:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

$d(i,j)$ é a razão entre os atributos discordantes pelo total de atributos.

2.1.4.2. ATRIBUTOS BINÁRIOS ASSIMÉTRICOS

Um atributo booleano é assimétrico se o valor 0 ou 1 não possuir a mesma importância para a análise de grupos, ou seja, caso haja uma predominância entre em algum dos valores. A distância entre atributos binários simétricos é dada por [4]:

$$d(i,j) = \frac{r+s}{q+r+s}$$

$d(i,j)$ é a razão entre os atributos discordantes pelo total de atributos, exceto quando ambos os valores são iguais a 0.

2.1.5. SIMILARIDADE DE JACCARD

É possível medir a distância entre atributos binários com base na similaridade, e não na dissimilaridade, um exemplo desse método é o coeficiente de Jaccard, que é uma razão para comparar a similaridade entre objetos de um conjunto, a similaridade é definida pela fórmula [4]:

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

Já a fórmula da similaridade assimétrica, por exemplo, pode ser calculada como:

$$\text{sim}(i,j) = \frac{q}{q+r+s} = 1 - d(i,j) .$$

2.2. CLASSIFICAÇÃO

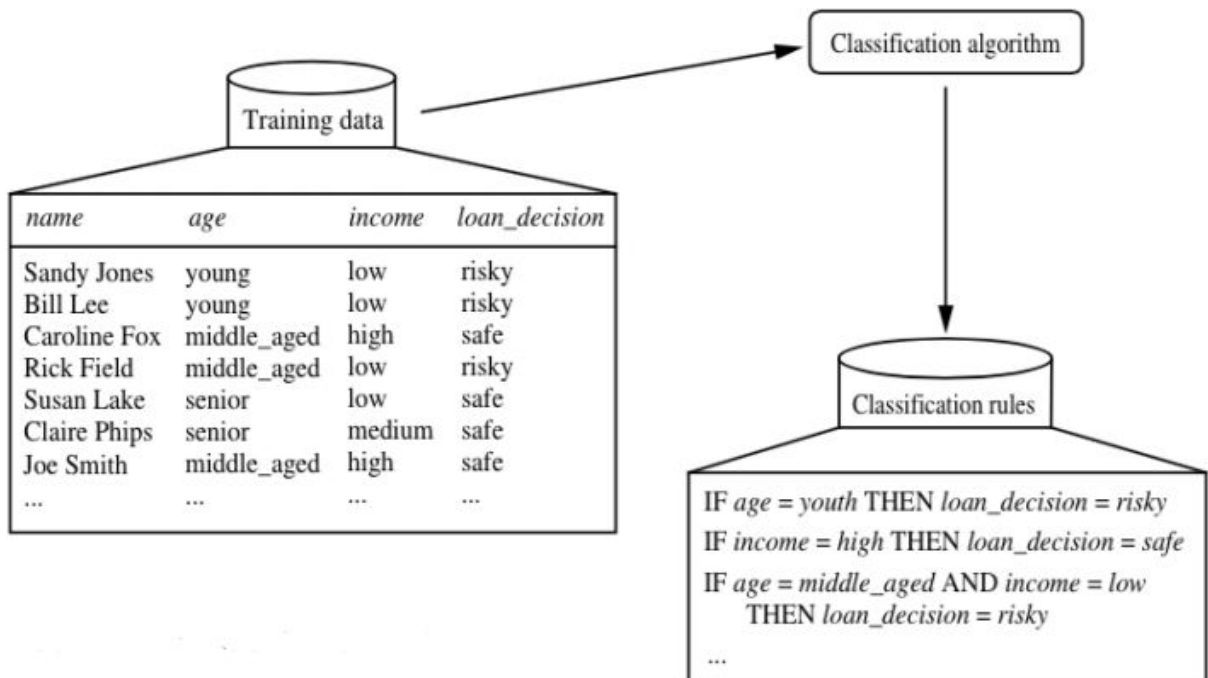
Através de registros na base de dados e um atributo classificador, que é o rótulo de classe, busca-se correlações entre os atributos, “o objetivo é construir um modelo que possa ser usado para prever qual seria essa saída para novos registros, ou seja, registros cuja classe ou valor de saída são desconhecidos.” (DE CASTRO; FERRARI, 2016, p. 148).

2.2.1. MODELO PREDITIVO

O modelo pode ser de diversas formas, como árvores de decisão, redes neurais, regras de classificação, entre outros. A capacidade de generalização é a capacidade que o modelo tem de gerar previsões satisfatórias, ou seja, o quão bom é a previsão do valor para novos registros não-rotulados.

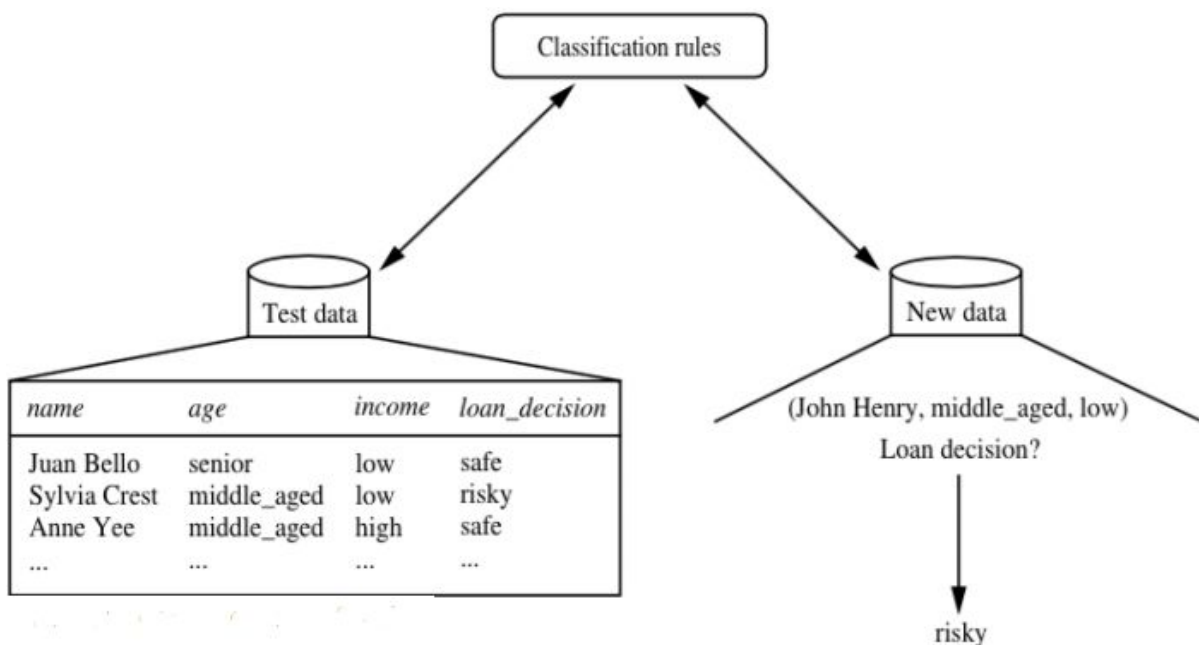
O seu desenvolvimento é separado em duas etapas, a etapa de treinamento e de teste, em que cada uma é formada por um subconjunto do conjunto total da base amostral, o subconjunto de treinamento é geralmente maior, podendo ser, por exemplo, 80% da base. O subconjunto de teste é menor, e possui, por exemplo, 20% das amostras da base.

- **Treinamento:** Nesta etapa, o preditor é gerado a partir do subconjunto de treinamento da base amostral, ele analisa a correlação entre os atributos e o valor do rótulo a fim de achar um modelo que de acordo com o conjunto de atributos, predite o valor da classe, que nesse subconjunto, é conhecido.

Figura 5 - Fase de treinamento

Fonte: Han & Kamber, 2006.

- **Teste:** Depois da geração do preditor, é preciso avaliar seu desempenho quando testado com dados que não foram utilizados na fase de treinamento.

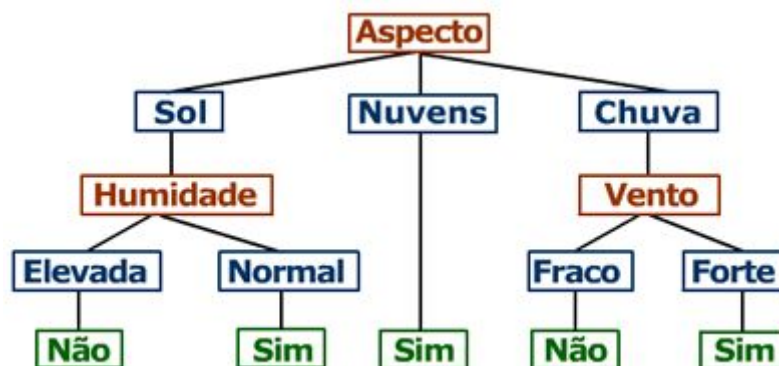
Figura 6 - Fase de teste

Fonte: Han & Kamber, 2006.

2.2.2. ÁRVORES DE DECISÃO

As árvores de decisão são estruturas construídas a partir de uma raiz, cada nó da árvore representa um teste sobre o atributo e cada ramo um resultado do teste, as folhas das árvores representam o valor das classes. O caminho da raiz até um nó folha corresponde a uma regra de classificação.

Figura 7 - Árvore de decisão



Fonte: GFBioInfo²

A construção de uma árvore de decisão pode ser feita de modo recursivo, feita em três passos:

- Selecione um atributo, coloque-o na raiz e a cada valor possível, crie uma ramificação na árvore;
- Repita esse processo recursivamente para cada ramo, usando apenas os objetos que alcançam esse ramo;
- Se todos os objetos de um nó possuem a mesma classificação, pare de desenvolver essa parte da árvore;

2.2.3. AVALIAÇÃO DE DESEMPENHO

As medidas de avaliação do desempenho trazem informações sobre a taxa de acerto ou de erro do classificador. A forma mais comum de avaliar um classificador é a partir da sua acurácia. O objetivo da avaliação é trazer um valor quantitativo da sua qualidade, e esse valor mostra quão bem o classificador classifica dados não usados no treinamento.

2.2.4. CLASSIFICAÇÃO BINÁRIA

Nos problemas de classificação binária existe uma classe alvo, ou seja, é a classe que se deseja prever. Essa classe alvo é a classe positiva, e o caso contrário, é a classe negativa. A partir dessas duas classes, são definidos tipos de medida específicas:

² Disponível em:

<<http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/indexf23d.html?id=199>>. Acesso em 05 jun. 2019

- **VP (verdadeiro positivo):** objeto da classe positiva classificado como positivo;
- **VN (verdadeiro negativo):** objeto da classe negativa classificado como negativo;
- **FP (falso positivo):** objeto da classe negativa classificado como positivo. Também é conhecido como Erro Tipo 1;
- **FN (falso negativo):** objeto da classe positiva classificado como negativo. Também é conhecido como Erro Tipo 2;

A partir dos valores das medidas apresentadas anteriormente, é possível apresentá-los a partir de uma matriz que relaciona as classes desejadas com as classes preditas. Essa matriz é chamada de matriz de confusão, matriz de contingência ou matriz de erro.

		Classe Predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Com base nos valores da matriz, é possível obter duas importantes taxas, a taxa de verdadeiros positivos(TVP) e a taxa de falsos positivos(TFP):

- **Taxa de verdadeiros positivos (TVP):** Corresponde ao percentual de objetos positivos classificados corretamente.

$$TVP = \frac{VP}{VP + FN}$$

- **Taxa de falsos positivos (TFP):** Corresponde ao percentual de objetos negativos classificados como positivos.

$$TFP = \frac{FP}{FP + VN}$$

A taxa global de sucesso do algoritmo pode ser calculada a parte da acurácia (ACC), ela é o número de classificações corretas pelo número total de classificações:

$$ACC = \frac{VP + VN}{VP + FP + VN + FN}$$

E a última taxa, é a taxa de erro (E), ela é calculada pela fórmula:

$$E = 1 - ACC$$

2.3. ASSOCIAÇÃO

Além das bases tradicionais, existem bases que são muito comuns em ambientes empresariais, elas armazenam transações, e por isso, cada registro é chamado de transação, e a base de transacional. Um exemplo típico dessa base é o carrinho de supermercado. Quando alguém vai ao supermercado e realiza uma compra, todos os itens comprados pelo cliente são

armazenados, os itens, a quantidade e os preços, tudo em um novo registro na base de dados do supermercado.

2.3.1. REGRAS DE ASSOCIAÇÃO

As regras de associação são obtidas a partir da análise da base, e podem ser encontradas relações entre os itens. A regra de associação é uma implicação na forma:

$$LHS \rightarrow RHS$$

LHS: Left-Hand-Side; *RHS*: Right-Hand-Side;

LHS e *RHS* são conjuntos que estão contidos na base e a intersecção entre eles é vazia.

As regras de associação podem ser vistas como padrões descritivos e representam a probabilidade de que um conjunto de itens apareça na relação, dado que o outro está presente, a regra acima é lida da seguinte maneira: “*LHS* implica *RHS*”

2.3.2. SUPORTE E CONFIANÇA

Uma grande variedade de regras de associação podem ser deduzidas de uma base transacional, para reduzi-las e selecionar somente as mais relevantes, duas métricas são utilizadas, o suporte e a confiança.

O suporte indica o número de transações na qual o conjunto de itens desejados aparece na base, ele é calculado a partir do número de ocorrências do conjunto na base pelo número total de transações. Formalmente, o suporte pode ser descrito pela fórmula:

$$suporte(X \rightarrow Y) = \frac{sup(X \rightarrow Y)}{|N|}$$

A confiança mede a confiabilidade da regra, ela indica a frequência com que o consequente da regra aparece em transações que contém o antecedente, é calculado pela seguinte fórmula:

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}$$

3. DESCRIÇÃO DO PROBLEMA

Um dos assuntos que está causando grande impacto e discussão na sociedade brasileira atual é a educação, muitas vezes os próprios governantes não fazem uma análise apropriada da conjuntura educacional, principalmente da educação superior. Nesse sentido nosso trabalho visa utilizar uma das ferramentas do KDD (Knowledge Discovery in Databases) que é a mineração de dados para limpar, analisar e retirar informações da base de dados do PROUNI 2018. Portanto a partir do dataset pretendemos examinar se é possível extrair insights concisos e possivelmente aplicáveis para o escopo do PROUNI. Em uma primeira abordagem foi fizemos uma classificação da base por meio de análise de

agrupamentos e em uma segunda abordagem definimos regras para a base com algoritmos de associação.

3.1. EXPLICAÇÃO DOS ATRIBUTOS

O dataset PROUNI 2018 contém 41447 instâncias (tuplas), sendo que cada uma dessas instâncias contém 16 atributos (ou *features*), dentro de 41447 instâncias apresenta-se um total de 29% de dados faltantes, outra característica é que a base não é rotulada, portanto as tuplas não tem classe.

- UF - Representa a unidade federativa que localiza-se a instituição que oferta as bolsas.
- Cidade - Representa o município que localiza-se a instituição que oferta as bolsas.
- Universidade - Nome da universidade que oferta as bolsas.
- Nome do Campus - Nome do campus da universidade que está o curso ofertado.
- Curso - Nome do curso da instituição que possui as vagas.
- Grau - Nível de instrução que o curso ofertado concede ao ser concluído.
- Turno - Período do dia o qual ocorre as aulas.
- Mensalidade - Preço integral do curso.
- Bolsas Integrais (cota) - Número de bolsas na modalidade Integral para cotistas.
- Bolsas Integrais (Ampla) - Número de bolsas na modalidade Integral para Ampla concorrência.
- Bolsas Parciais (cota) - Número de bolsas na modalidade Parcial para cotistas.
- Bolsas Parciais (Ampla) - Número de bolsas na modalidade Parcial para Ampla concorrência.
- Nota Integral (cota) - Nota no enem mínima para a bolsa na modalidade de Bolsa Integral como cotista.
- Nota Integral (Ampla) - Nota no enem mínima para a bolsa na modalidade de Bolsa Integral como ampla concorrência.
- Nota Parcial (Ampla) - Nota no enem mínima para a bolsa na modalidade de Bolsa parcial ampla concorrência.
- Nota Parcial (cota) - Nota no enem mínima para a bolsa na modalidade de bolsa parcial.

Podemos ter uma visão geral da base utilizando o nó *file* do *Orange Canvas*, perceba que na coluna *Name* temos o nome de cada atributo e na coluna *Type* temos o tipo de cada atributo, para uma visualização dos atributos a coluna *Role* não nos interessa no momento.

Figura 8 - Visualização dos atributos da base antes do pré-processamento

The screenshot shows a data visualization application window. The 'File' menu is open, showing 'cursos.csv' as the selected file. The 'Info' section, highlighted with a red box, displays the following information:

- 41447 instance(s)
- 15 feature(s) (29.0% missing values)
- Data has no target variable.
- 1 meta attribute(s)

The 'Columns' section, also highlighted with a red box, displays a table with 16 columns. The table has the following structure:

	Name	Type	Role	Values
1	uf_busca	Categorical	feature	AC, AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, PA, PB, PE, PI, PR, RJ, RN, ...
2	cidade_busca	Categorical	skip	Abaete, Abaetetuba, Abaira, Abelardo Luz, Acailandia, Acarau, Acrelandia, Acu, Adamantina, Afogados da Ingazeira, Afonso Claudio, Agrolandia, Agua Boa, Agua Branca, Aguai, Aguas Belas, Aguas Lindas de Goias, Agu...
3	universidade_n...	Categorical	skip	Alfa - Faculdade de Almenara, Atopp Brasil Faculdade de Negocios - Atopp Brasil - ATOPP BRASIL, Centro Superior de Estudos Juridicos Carlos Drummond de Andrade - Csejca - CSEJCDA, Centro Técnico-Educacional ...
4	nome	Categorical	meta	Administração, Administração Pública, Administração de Empresas, Administração de Recursos Humanos, Agroindústria, Agronegócio, Agronomia, Agropecuária, Alimentos, Análise e Desenvolvimento de Sistemas, Arq...
5	grau	Categorical	feature	Bacharelado, Licenciatura, Tecnológico
6	turno	Categorical	feature	Curso a Distância, Integral, Matutino, Noturno, Vespertino
7	mensalidade	Numeric	feature	
8	bolsa_integral_c...	Categorical	feature	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, ...
9	bolsa_integral_a...	Categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
10	bolsa_parcial_c...	Categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
11	bolsa_parcial_a...	Categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
12	nota_integral_a...	Numeric	feature	
13	nota_integral_c...	Numeric	feature	
14	nota_parcial_a...	Numeric	feature	
15	nota_parcial_co...	Numeric	feature	
16	campus_nome	Text	skip	

The 'Info' section and the 'Columns' table are highlighted with red boxes in the original image. The 'Columns' table is also highlighted with a red box. The 'Info' section is also highlighted with a red box. The 'Columns' table is also highlighted with a red box. The 'Info' section is also highlighted with a red box. The 'Columns' table is also highlighted with a red box.

Fonte: print screen da aplicação no sistema operacional Windows 10.

Para uma visualização mais apurada da situação de cada atributos podemos utilizar o nó *feature statistics*, onde temos uma medida de tendência central e dispersão para cada atributo e também uma visualização da porcentagem individual de valores faltantes. Observe que os

atributos *nota_parcial_cotas* e *bolsa_parcial_cotas* são os atributos com maior quantidade de dados faltantes.

Figura 9 - Visualização das estatísticas por atributo



Fonte: print screen da aplicação no sistema operacional Windows 10.

3.2. EXPLICAÇÃO DO PRÉ-PROCESSAMENTO

3.2.1. PRÉ-PROCESSAMENTO PARA TAREFA DE ASSOCIAÇÃO

A fase de pré-processamento da base era pré-requisito para a construção da associação dos dados, uma vez que a tarefa de associação via os softwares utilizados requer uma base transacional. Nossa base era composta inicialmente por strings e atributos números que obedeciam um determinado range. Então, a base original foi totalmente transacionada, eliminando certos atributos e criando novos. A fase de pré-processamento para associação consistiu em:

- Analisar o dataset
- Escolher atributos relevantes
- Exclusão de atributos “irrelevantes”
- Criar novos atributos
- Transaciona os atributos restantes
- Exclusão dos atributos originais restantes que já foram transacionados.

3.2.2. PRÉ-PROCESSAMENTO PARA TAREFA DE AGRUPAMENTO

A fase de pré-processamento dos dados pode ser considerada a fase mais crucial de um processo de mineração de dados [5], devido ao nosso trabalho ter gerado dois artefatos,

um clustering e uma associação da base, nosso pré-processamento deve de ser dividido em duas partes, uma de pré-processamento para o clustering e uma de pré-processamento para a associação. A fase de pré-processamento para o clustering consistiu em:

- Analisar o dataset
- Escolher atributos relevantes
- Imputar dados faltantes
- Discretização contínua de *categoricals*
- Transformação do dataset

3.3. EXPLICAÇÃO DA ESCOLHA DA TAREFA PARA O PROBLEMA

3.3.1. AGRUPAMENTO

A escolha do agrupamento como tarefa para nossa base é dada pela falta de informações que temos sobre ela, é uma base com muitos dados e com muito potencial, porém é uma base em rótulo e não temos nenhuma informação a mais sobre ela além dos dados contidos. Portanto, o agrupamento irá nos ajudar a encontrar relações entre os dados contidos na base, assim extraíndo grupos e posteriormente rotulando a base se possível, ademais, mesmo que nenhuma rotulação óbvia apareça em um primeiro momento, o clustering nos dará informações, para que com isso possamos construir conhecimento sobre a base e seguir com outras tarefas, visto isso escolhemos o algoritmo *k-means* para realizar nosso agrupamento, pois é um algoritmo visto em sala de aula e também confiável e simples.

3.3.2. CLASSIFICAÇÃO

A tarefa de agrupamento é crucial para entendimento das relações dos dados do *dataset*, porém, após o agrupamento podemos ter *insights* que nos gerem rótulos à base, dito isso utilizamos a classificação para criar modelos de predição que executam em cima da base rotulada gerada pós-agrupamento, assim com amostras de treinamento e de teste podemos ver como os algoritmos de classificação se comportam classificando as tuplas com os rótulos criados por nós. De um modo geral isso servirá até para *datasets* futuros do PROUNI, visto que se extrapolarmos as características encontradas aqui podemos utilizar nosso treinamento para classificar uma base futura.

3.3.3. ASSOCIAÇÃO

A dificuldade de encontrar agrupamentos nos levou a procurar outras meios de mineração de dados na base, acarretando na escolha da associação como uma das tarefas aplicadas a ser aplicada na base. Dessa forma, viu-se a possibilidade de elaborar uma matriz de contingência dos dados para calcular a distância entre atributos binários, porém, não foi possível fazer isso via software com uma base transacional, entretanto, uma vez que a base é transacionada é possível verificar associações na base utilizando o software Weka e o algoritmo *a priori*. Portanto, as associações na base irá nos ajudar a encontrar certos padrões presentes no PROUNI, mesmo que sejam rotulações com alto nível de suporte e confiança,

elas não são associações triviais de se notar, uma vez que pouco se conhece sobre os critérios internos das faculdade para oferecimento de bolsas.

4. METODOLOGIA

4.1. PRÉ-PROCESSAMENTO

4.1.1. PREPARAÇÃO PARA AGRUPAMENTO

Começando pelo clustering, utilizamos uma abordagem inicial de análise das features, visando compreender melhor nossa base e selecionar os atributos que possivelmente são mais relevantes, para tal começamos visualizando os atributos e descartando alguns atributos que provavelmente iriam atrapalhar nosso algoritmo de clustering, esses sendo: {nome, cidade_busca, universidade_nome, campus_nome}. O fato desses atributos serem categóricos e terem muitas categorias iria “fazer mal” para nosso clustering, poderíamos tratar esses dados de outra maneira, porém pelo tempo e objetivo deste trabalho preferimos seguir com a base da seguinte maneira:

Figura 10 - Feature selection para clustering da base

	Name	Type	Role	Values
1	uf_busca	C categorical	feature	AC, AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, PA, PB, PE, PI, PR, RJ, RN
2	cidade_busca	C categorical	skip	Abaete, Abaetetuba, Abaira, Abelardo Luz, Acailandia, Acarau, Acrelandia, Acu, Adamantina, Afogados da Ingazeira, Afonso Claud...
3	universidade_n...	C categorical	skip	Alfa - Faculdade de Almenara, Atopp Brasil Faculdade de Negócios - Atopp Brasil - ATOPP BRASIL, Centro Superior de Estudos Jurídicos ...
4	nome	C categorical	skip	Administração, Administração Pública, Administração de Empresas, Administração de Recursos Humanos, Agroindústria, Agronegócio, ...
5	grau	C categorical	feature	Bacharelado, Licenciatura, Tecnológico
6	turno	C categorical	feature	Curso a Distância, Integral, Matutino, Noturno, Vespertino
7	mensalidade	N numeric	feature	
8	bolsa_integral_c...	C categorical	feature	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, ...
9	bolsa_integral_a...	C categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
10	bolsa_parcial_c...	C categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
11	bolsa_parcial_a...	C categorical	feature	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
12	nota_integral_a...	N numeric	feature	
13	nota_integral_c...	N numeric	feature	
14	nota_parcial_a...	N numeric	feature	
15	nota_parcial_co...	N numeric	feature	
16	campus_nome	S text	skip	

Fonte: print screen da aplicação no sistema operacional Windows 10.

Observe que os atributos dentro dos retângulos verdes foram mantidos e os dentro dos retângulos vermelhos foram atribuídos como *skip*, assim, serão ignorados pelos algoritmos. Veja como a base se figura agora que selecionamos os atributos que iremos trabalhar:

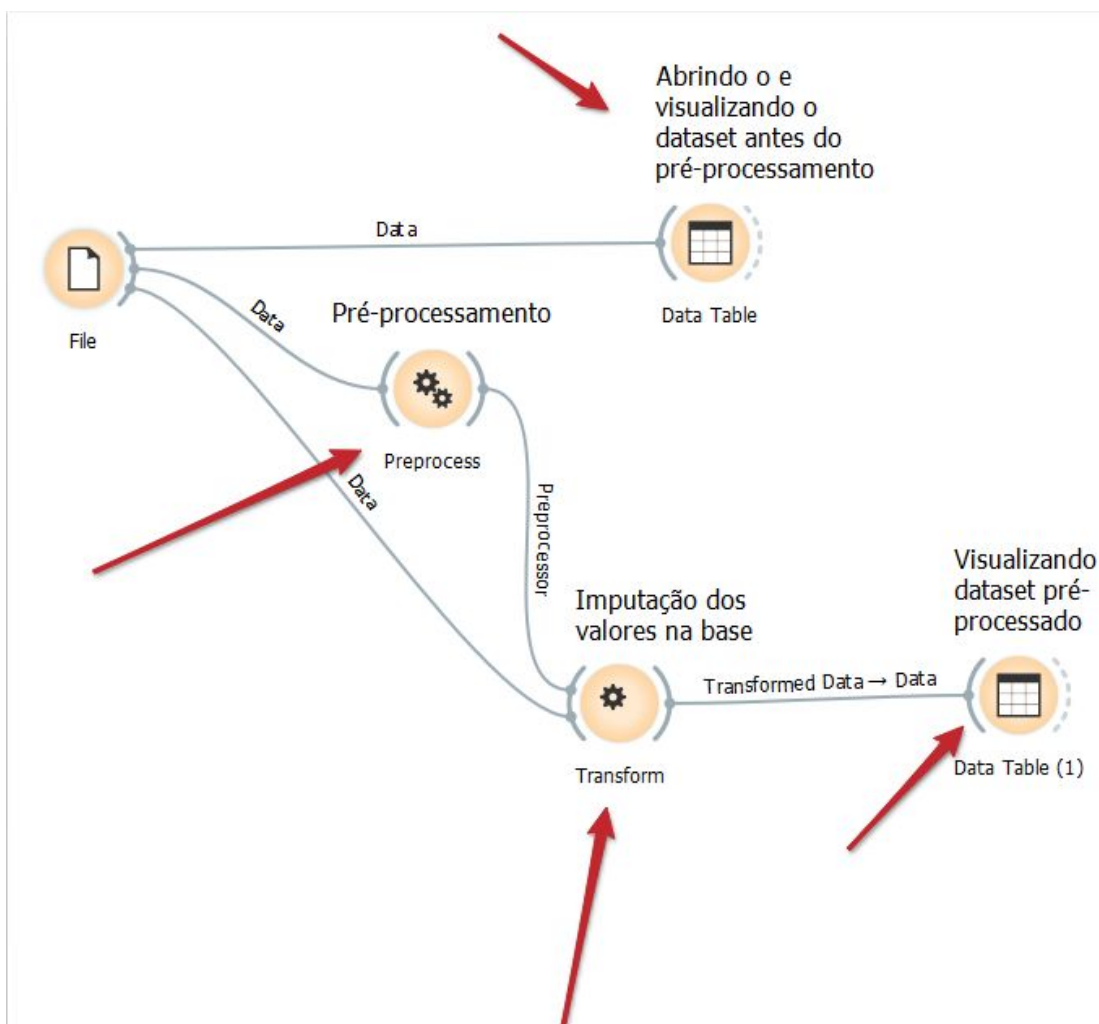
Figura 11 - Head da base com atributos escolhidos

	uf_busca	grau	turno	mensalidade	olsa_integral_cota	olsa_integral_ampl	olsa_parcial_cota	olsa_parcial_ampl	ota_integral_ampl	vota_integral_cota	vota_parcial_ampl	nota_parcial_cota
1	AC	Bacharelado	Curso a Distância	289.00	1	1	?	?	572.74	548.00	?	?
2	AC	Bacharelado	Curso a Distância	298.00	1	?	?	?	646.14	?	?	?
3	AC	Bacharelado	Curso a Distância	325.00	?	?	1	?	?	?	577.62	?
4	AC	Bacharelado	Curso a Distância	319.00	1	?	?	?	616.68	?	?	?
5	AC	Bacharelado	Curso a Distância	298.00	?	?	1	?	?	?	569.00	?
6	AC	Bacharelado	Noturno	823.22	?	?	?	2	?	?	564.24	?
7	AC	Bacharelado	Vespertino	476.00	4	1	6	2	612.24	595.18	569.04	555.28
8	AC	Bacharelado	Noturno	476.00	1	1	4	1	608.18	607.58	585.72	565.58
9	AC	Bacharelado	Curso a Distância	325.00	?	?	1	?	?	?	562.98	?
10	AC	Bacharelado	Noturno	522.79	0	2	?	2	606.40	?	582.24	?

Fonte: print screen da aplicação no sistema operacional Windows 7.

Agora precisamos imputar os dados faltantes na nossa base, utilizando os nós de pré-processamento da ferramenta *Orange Canvas 3*, realizamos um workflow onde a base é inserida, pré-processada, transformada e visualizada.

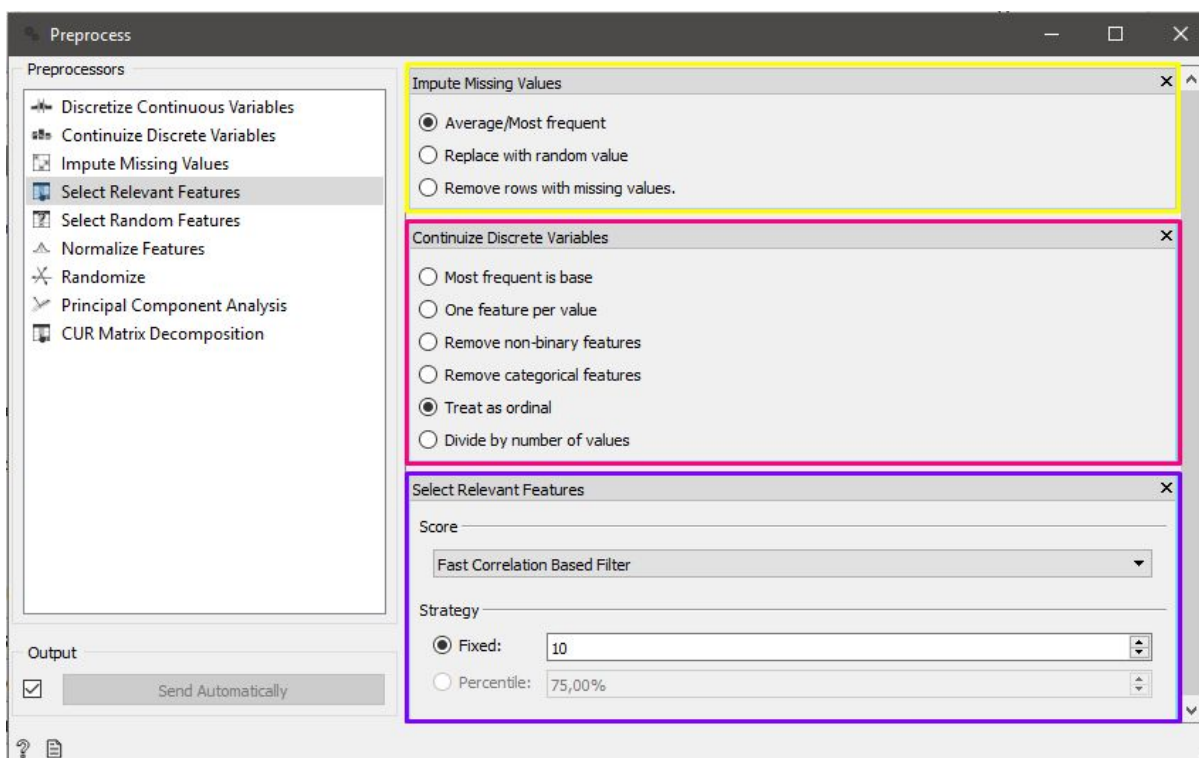
Figura 12 - Workflow de pré-processamento no Orange Canvas 3



Fonte: print screen da aplicação no sistema operacional Windows 7.

Visto o workflow, podemos começar visualizando quais tarefas foram utilizadas no pré-processamento, na imagem abaixo temos em amarelo a imputação dos valores faltantes por meio de medidas de tendência central, média/moda. No retângulo rosa temos a discretização contínua de variáveis discretas, isto é, os atributos de tipo *categorical* como UF_busca se tornaram valores numéricos, para cada categoria foi atribuído um número, por exemplo para Acre -> 1, para Amapá -> 2 e assim por diante. Isto foi realizado para “satisfazer” melhor os algoritmos de clustering. Em roxo temos a seleção dos atributos relevantes por meio da correlação, selecionamos um valor fixo de 10 atributos, sendo que a base no momento contém 12, para que o *Orange* automaticamente selecione as 10 *features* mais importantes baseando-se em suas correlações.

Figura 13 - Tratamento da base com o nó *preprocess* do *Orange Canvas*



Fonte: print screen da aplicação no sistema operacional Windows 10.

Agora que tratamos nossa base podemos ter uma visualização do artefato gerado pelos algoritmos, esse é o último passo antes de partimos para as tarefas de mineração propriamente ditas, devemos observar e determinar se a base está pronta ou não para ser minerada.

Figura 14 - Base transformada após pré-processamento

	uf_busca	grau	turno	mensalidade	olsa_integral_cota	olsa_integral_amp	olsa_parcial_cota	olsa_parcial_amp	ota_integral_amp	vota_integral_cota
1	0.000	0.000	0.000	289.00	1.000	0.000	0.000	0.000	572.74	548.00
2	0.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	646.14	564.47
3	0.000	0.000	0.000	325.00	1.000	0.000	0.000	0.000	575.98	564.47
4	0.000	0.000	0.000	319.00	1.000	0.000	0.000	0.000	616.68	564.47
5	0.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	575.98	564.47
6	0.000	0.000	3.000	823.22	1.000	0.000	0.000	1.000	575.98	564.47
7	0.000	0.000	4.000	476.00	4.000	0.000	5.000	1.000	612.24	595.18
8	0.000	0.000	3.000	476.00	1.000	0.000	3.000	0.000	608.18	607.58
9	0.000	0.000	0.000	325.00	1.000	0.000	0.000	0.000	575.98	564.47
10	0.000	0.000	3.000	522.79	0.000	1.000	1.000	1.000	606.40	564.47
11	0.000	0.000	3.000	672.15	2.000	0.000	0.000	0.000	567.98	568.88
12	0.000	0.000	0.000	250.00	1.000	0.000	0.000	0.000	575.98	564.47
13	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	575.98	564.47
14	1.000	0.000	3.000	586.63	2.000	0.000	0.000	0.000	658.26	593.30
15	1.000	0.000	0.000	290.35	1.000	0.000	0.000	0.000	609.46	583.92
16	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	602.32	564.47
17	1.000	0.000	0.000	417.14	2.000	0.000	0.000	0.000	534.22	569.72
18	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	573.20	564.47
19	1.000	0.000	3.000	586.63	2.000	0.000	0.000	0.000	598.60	578.98
20	1.000	0.000	2.000	669.00	3.000	1.000	0.000	0.000	581.96	590.30
21	1.000	0.000	3.000	669.00	7.000	2.000	0.000	0.000	602.46	577.94
22	1.000	0.000	3.000	169.00	10.000	4.000	0.000	0.000	573.90	562.80
23	1.000	0.000	3.000	464.31	2.000	0.000	0.000	0.000	631.96	574.86
24	1.000	0.000	3.000	535.00	3.000	1.000	0.000	0.000	580.46	580.48
25	1.000	0.000	3.000	712.86	5.000	2.000	0.000	0.000	583.18	564.66
26	1.000	0.000	2.000	627.14	5.000	1.000	0.000	0.000	571.32	583.60
27	1.000	0.000	3.000	357.00	1.000	0.000	2.000	10.000	575.98	564.47
28	1.000	0.000	3.000	548.00	7.000	2.000	0.000	0.000	577.14	567.40
29	1.000	0.000	2.000	366.90	2.000	0.000	0.000	0.000	566.40	556.98

Fonte: print screen da aplicação no sistema operacional Windows 10.

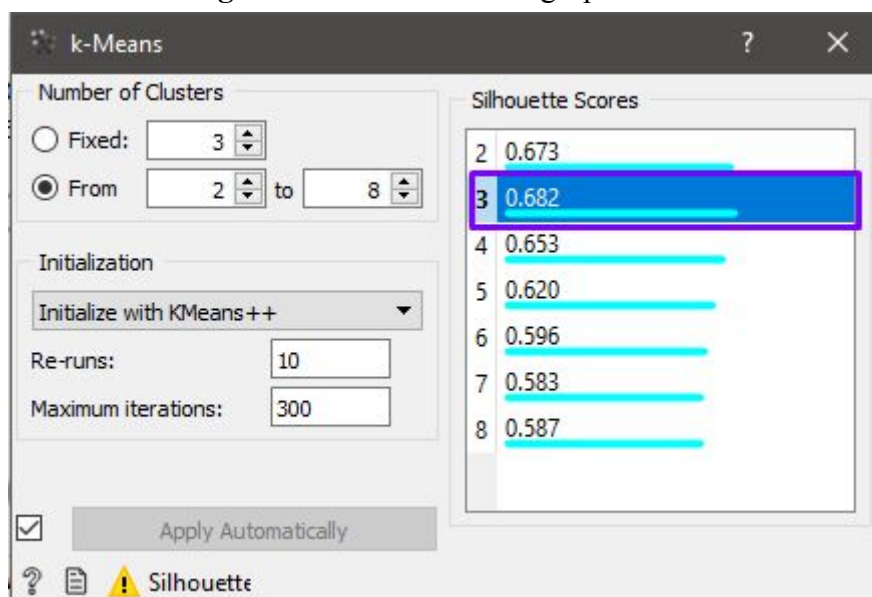
Visto isso, terminamos nossa fase de pré-processamento para a tarefa de clustering, a seguir temos o pré-processamento para a tarefa de associação.

4.2. TAREFA DE MINERAÇÃO

4.2.1. AGRUPAMENTO

Depois do pré-processamento foi feito um agrupamento utilizando-se da ferramenta *Orange Canvas 3* e do algoritmo *k-means*, foi realizado um agrupamento de 2 a 8 grupos, escolhemos o agrupamento com o melhor coeficiente de silhueta entre os 6, que foi o agrupamento com $K = 3$.

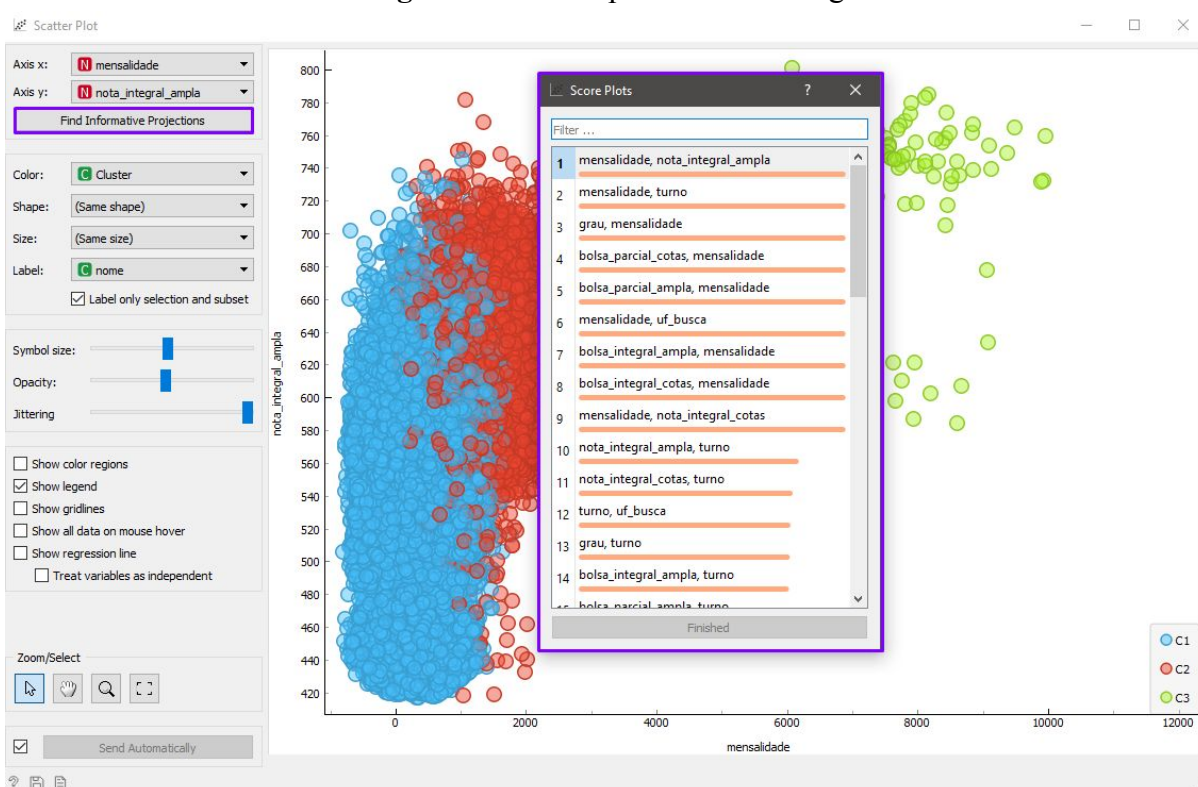
Figura 15 - Silhuetas dos agrupamentos



Fonte: print screen da aplicação no sistema operacional Windows 10.

Após realizado o agrupamento utilizamos a ferramenta de *score test* do Orange para que ele nos mostrasse as projeções que forneceriam informações no plot dos grupos em um plano R^2 .

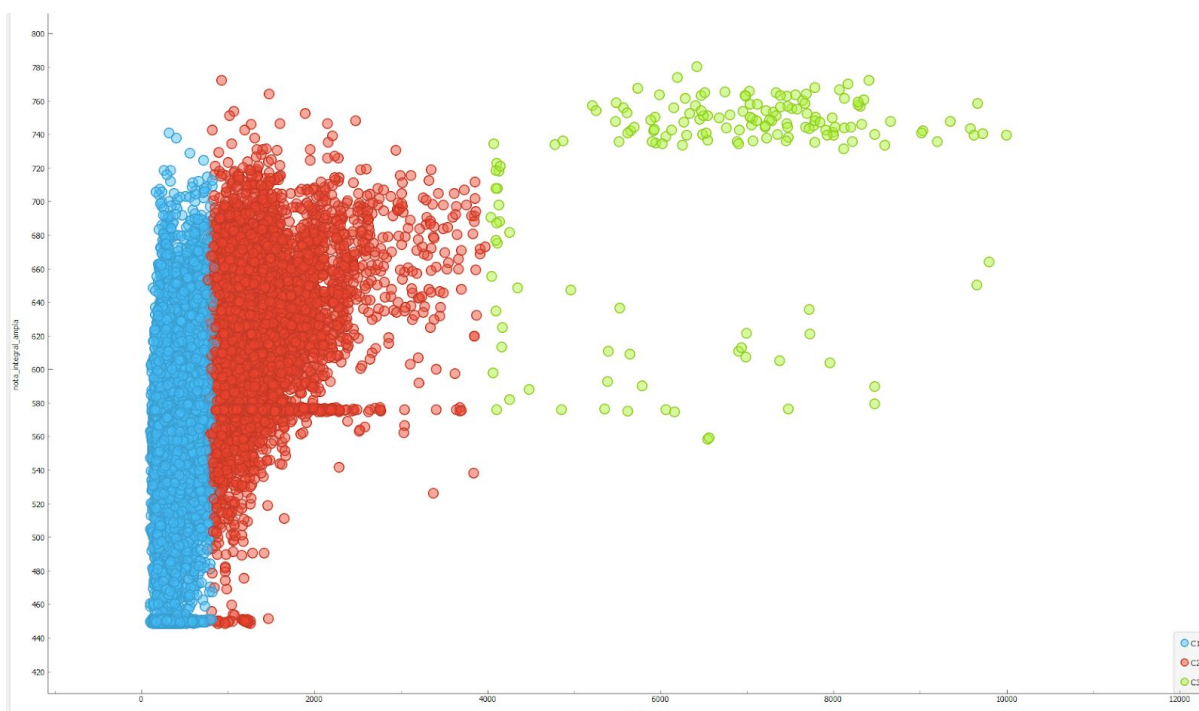
Figura 16 - Score plots do clustering



Fonte: print screen da aplicação no sistema operacional Windows 10.

Visto os score plots, podemos analisar e visualizar os plots que mais nos trazem informações sobre a base, para começar temos o plot de mensalidade x nota_integral_ampla, esse plot mostra 3 grupos, o grupo 3 representado pelo verde, o grupo 2 representado pelo vermelho e o grupo 1 representado pelo verde, uma das características mais marcantes deste plot é que o grupo 3 fica bem separado dos demais, o grupo 2 e 1 tem uma separação concisa apesar de pouco demarcada.

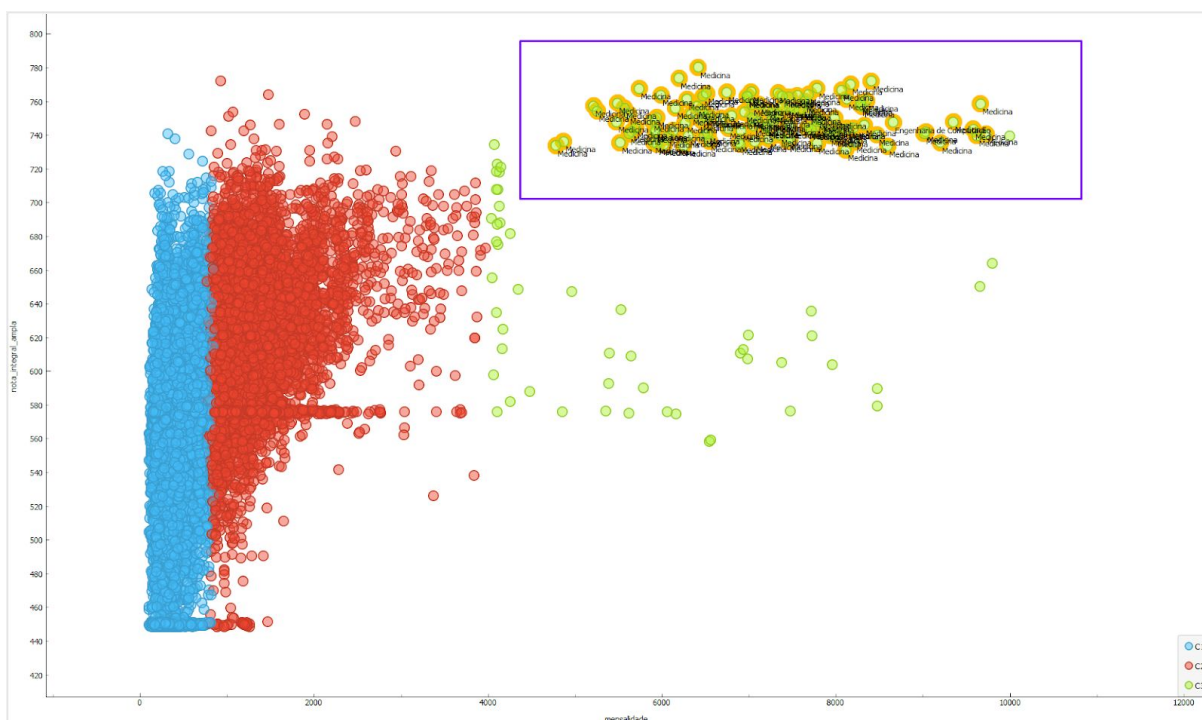
Figura 17 - Plot mensalidade x nota_integral_ampla



Fonte: print screen da aplicação no sistema operacional Windows 10.

Uma das coisas que salta aos olhos nesse plot é a grande separação do grupo verde para os demais, apesar das 44 mil tuplas no plot ele continua tendo uma boa separação, se demarcarmos o “label” de cada ponto para o nome do curso, podemos ver que a maioria dos cursos no grupo verde são cursos que tem maior concorrência no *prouni*, sendo que a maioria deles são cursos de medicina.

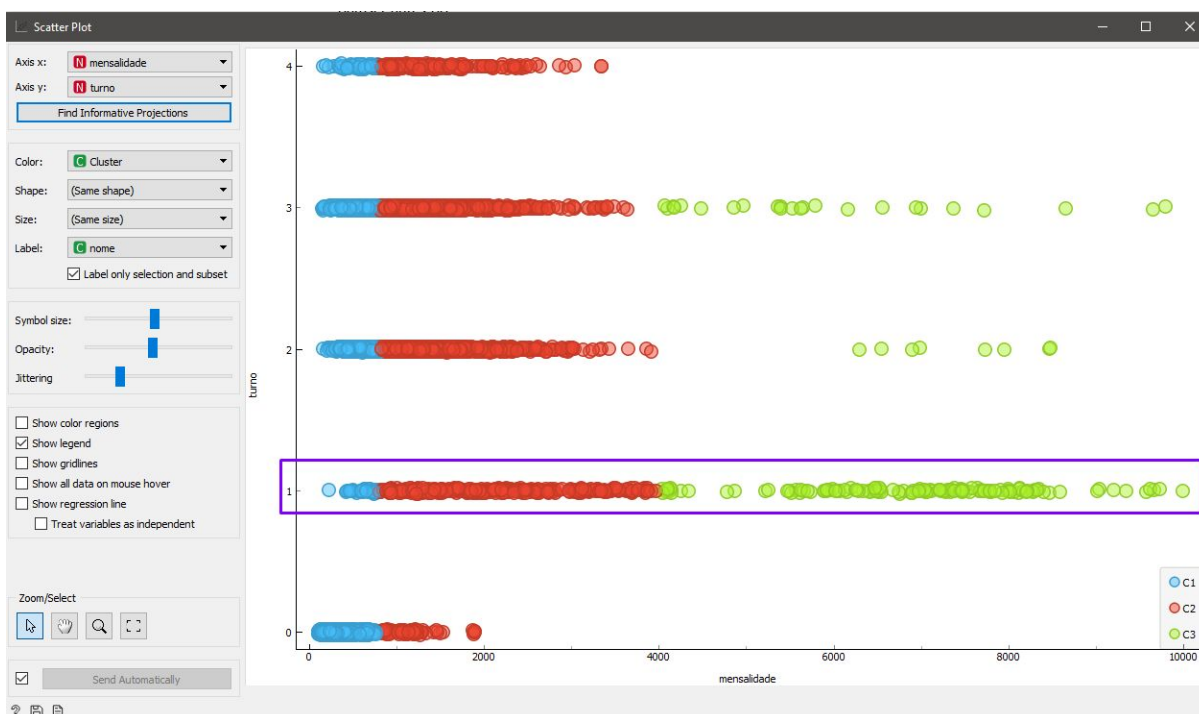
Figura 18 - Cursos concorridos no grupo 3



Fonte: print screen da aplicação no sistema operacional Windows 10.

No segundo plot temos a mensalidade pelo turno, como principal informação podemos extrair que os cursos integrais, sendo que $1 = INTEGRAL$, tem muito mais cursos do grupo 3, mostrando mais uma vez que o grupo 3 está atrelado a mensalidade e agora também mostrando que está atrelado a cursos integrais, que costumam ser Engenharias, cursos de saúde e etc.

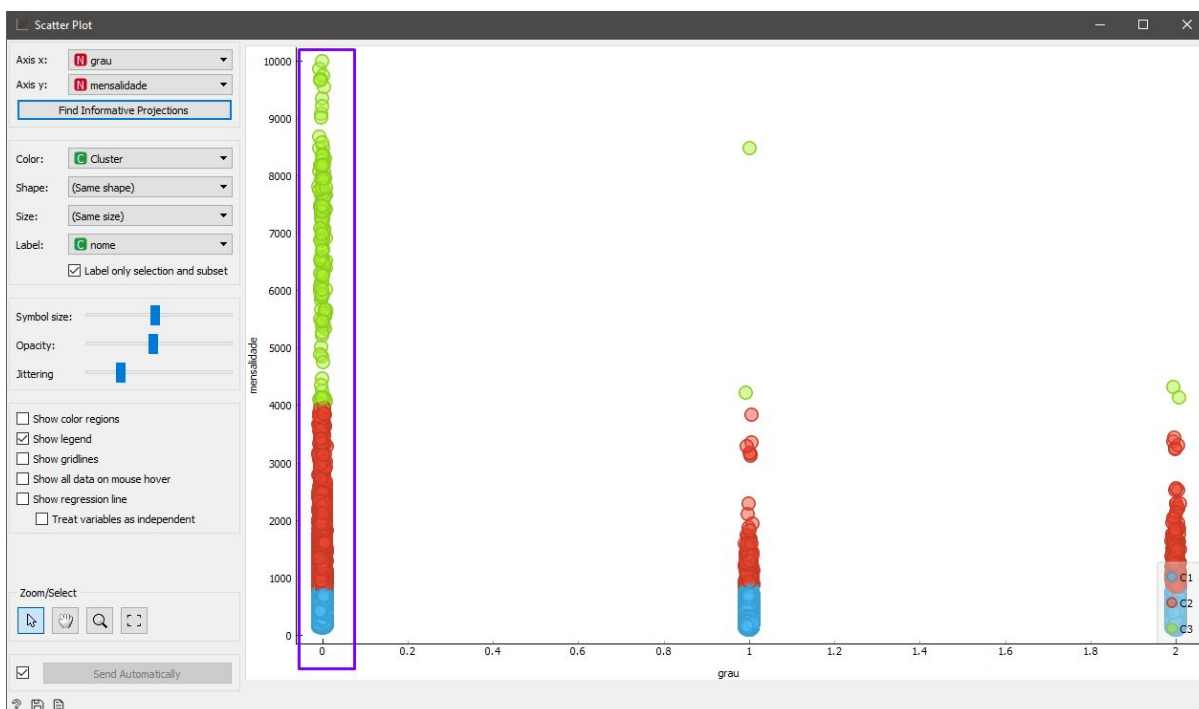
Figura 19 - Plot de mensalidade por turno



Fonte: print screen da aplicação no sistema operacional Windows 10.

Em um terceiro plot temos o grau pela mensalidade, mostrando que cursos bacharel, sendo que na discretização contínua temos que 1 = *BACHAREL*, tem praticamente todo os cursos do grupo 3, e está muito atrelado a mensalidade, até aqui podemos inferir que cursos com alta procura [6] estão bem presentes no cluster 3.

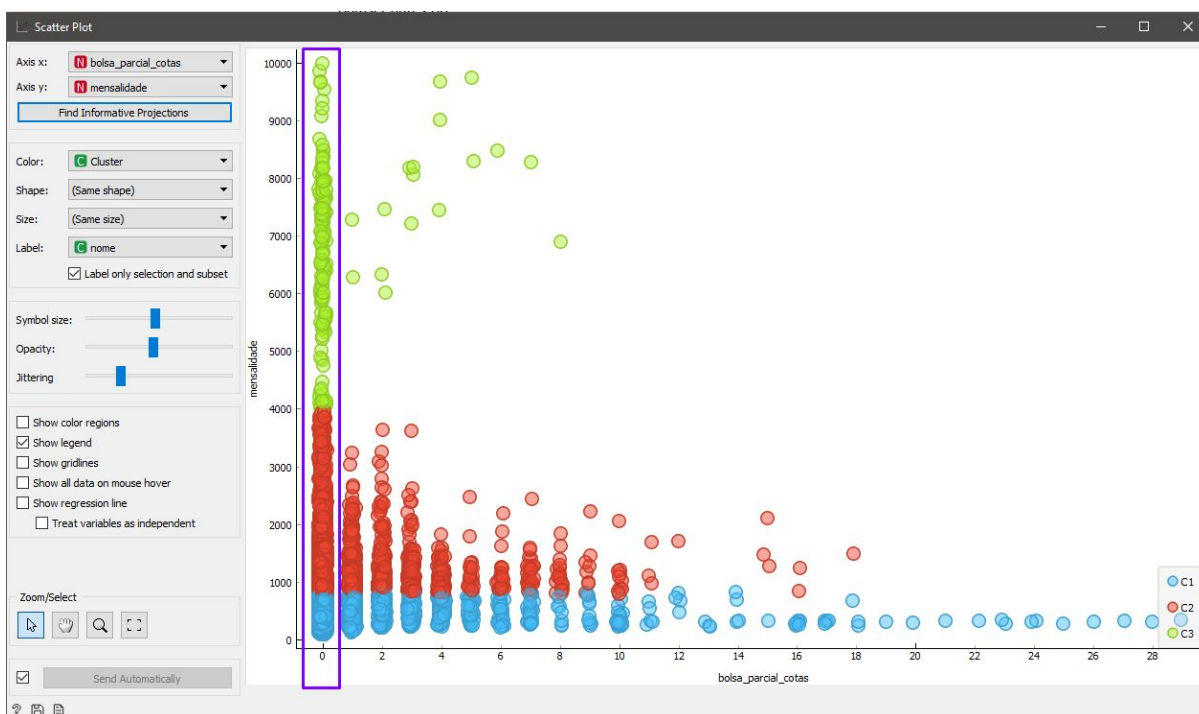
Figura 20 - Grau pela mensalidade



Fonte: print screen da aplicação no sistema operacional Windows 10.

Agora nos plots de mensalidade x bolsa, tanto parcial quando integral ampla e cotas temos um comportamento presente, os cursos do cluster 3 praticamente não oferecem bolsas, o que nos leva a crer que são cursos de alta demanda, já que as faculdade teoricamente não iriam precisar oferecer tantas bolsas no *prouni* pois a demanda de alunos supre a oferta do curso.

Figura 21 - *plot* da bola_parcial_cotas pela mensalidade



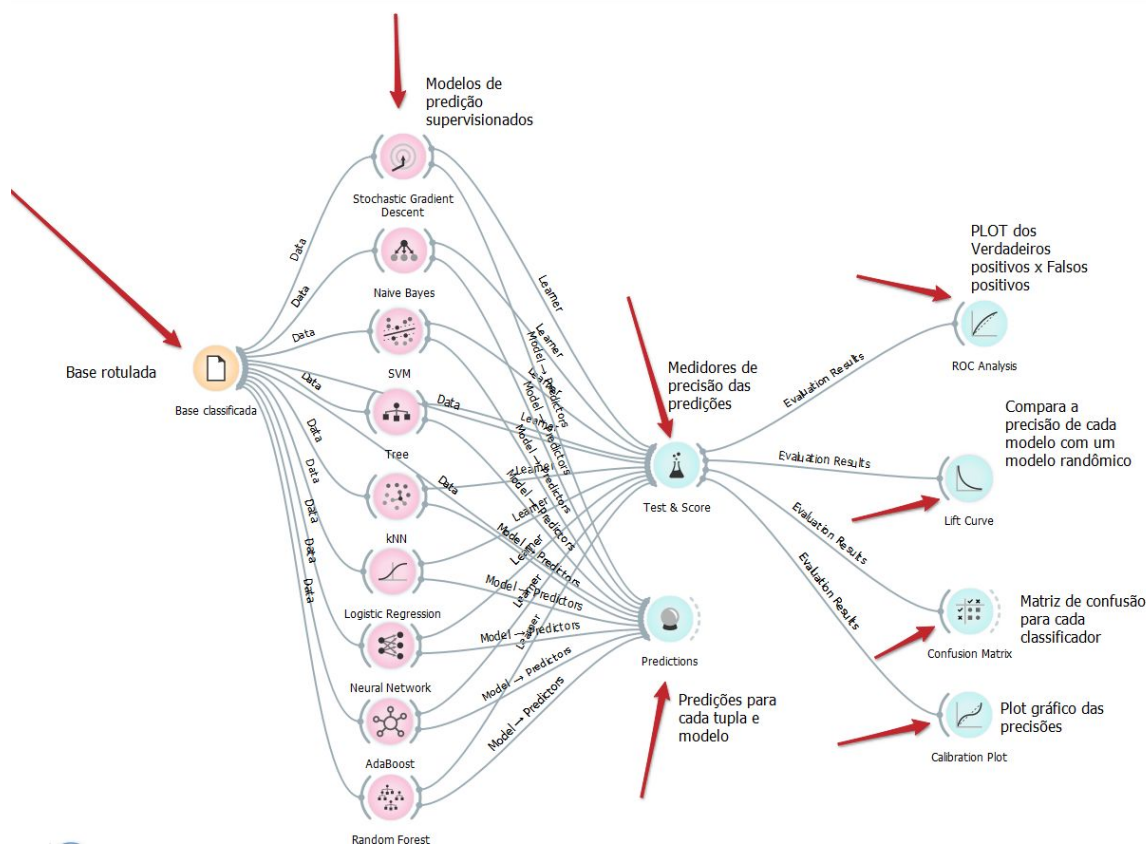
Fonte: print screen da aplicação no sistema operacional Windows 10.

4.2.2. CLASSIFICAÇÃO

Após a rotulação da base, abrimos as portas para modelos supervisionados, assim criamos uma rotina de classificação utilizando diversos preditores para a nossa base rotulada. Observe que utilizamos os algoritmos:

- Stochastic Gradient Descent
- Naive Bayes
- SVM
- Tree
- kNN
- Logistic Regression
- Neural Network
- AdaBoost
- Random Forest

Figura 22 - Rotina de classificação



Fonte: print screen da aplicação no sistema operacional Windows 10.

4.2.3. ASSOCIAÇÃO

4.2.3.1. PREPARAÇÃO PARA ASSOCIAÇÃO

Começou pela escolha inicial de quais atributos ficariam e quais seriam excluídos da base para transformação da base. A escolha dos atributos foi feita se baseando no propósito do PROUNI que é a seleção de bolsas em diversas modalidades para cursos de graduação, bacharelado, tecnológico regulamentados pelo MEC, não foi possível discretizar os seguintes atributos da base:

Nome do curso - Na base existem um total de 296 cursos, isso adicionaria na base a quantidade de 296×41476 , não sendo possível de associarmos tantos dados com o poder computacional disponível do grupo.

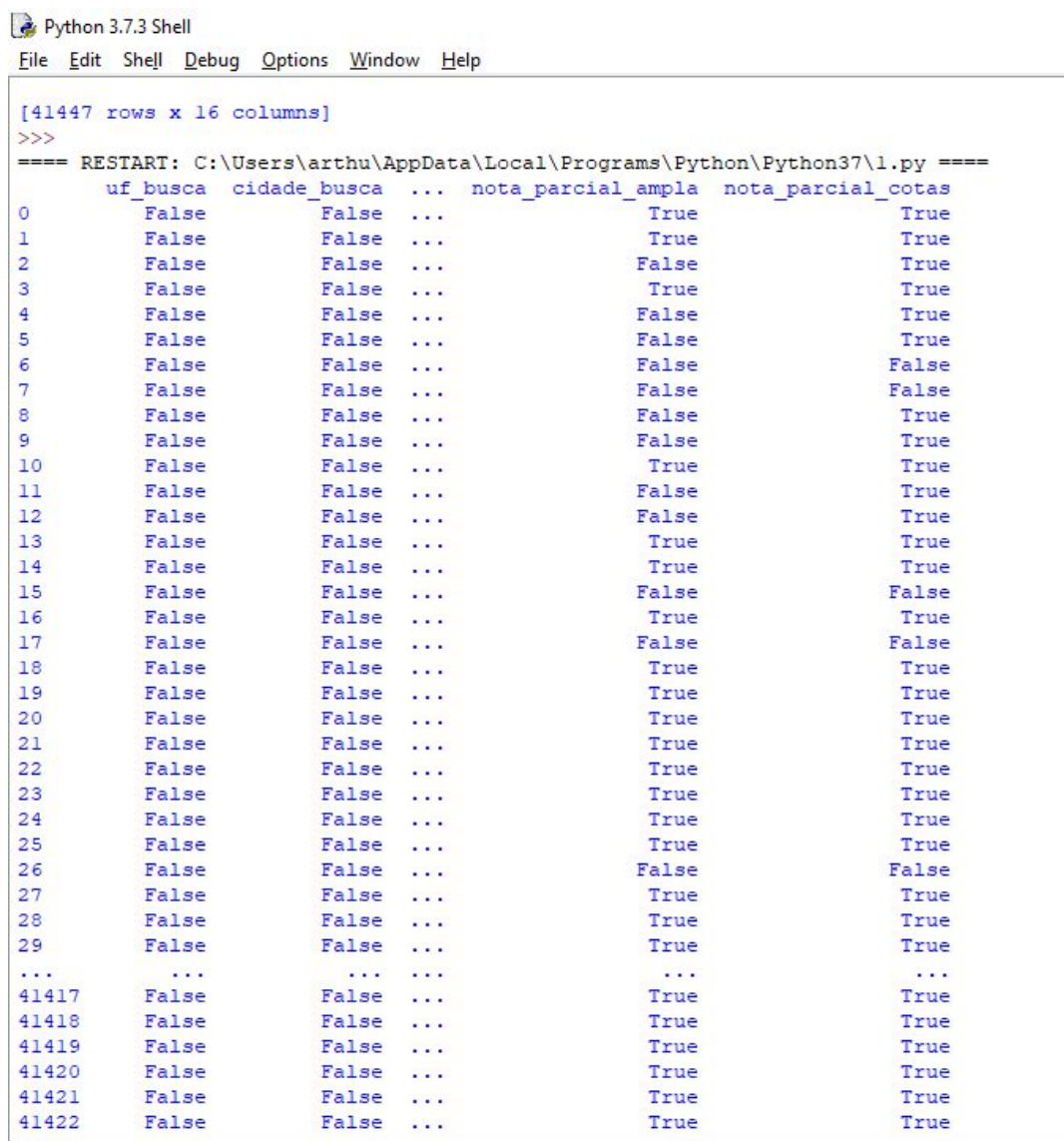
Estado, nome da universidade, campus, não foram transacionados pelo mesmo motivo acima.

4.2.3.2. COMO FOI FEITO?

O preparo da base foi feito inicialmente em um algoritmo montado em python, chegando a resultados como o da imagem abaixo, entretanto, viu-se uma melhor opção

utilizar o excel e vba para transformar a base, devido espaço de tempo curto para aprofundar nos conhecimentos de python.

Figura 23 - Base transacional em Python.



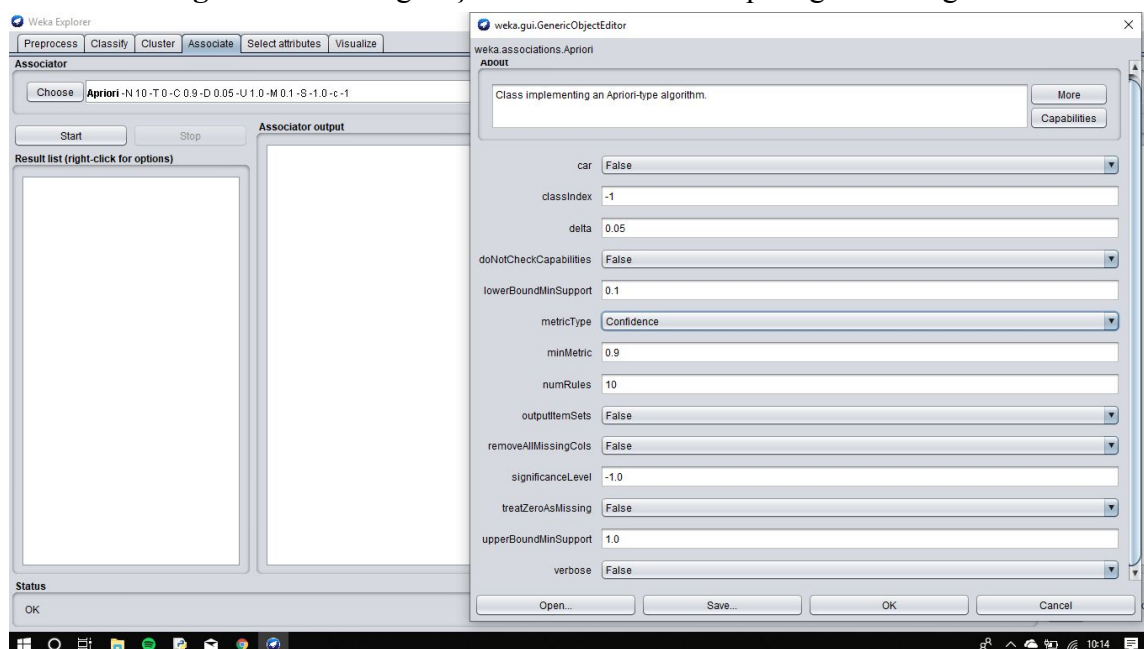
```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

[41447 rows x 16 columns]
>>>
===== RESTART: C:\Users\arthur\AppData\Local\Programs\Python\Python37\1.py =====
uf_busca  cidade_busca  ...  nota_parcial_ampla  nota_parcial_cotas
0         False      False  ...                True                True
1         False      False  ...                True                True
2         False      False  ...                False               True
3         False      False  ...                True                True
4         False      False  ...                False               True
5         False      False  ...                False               True
6         False      False  ...                False               False
7         False      False  ...                False               False
8         False      False  ...                False               True
9         False      False  ...                False               True
10        False      False  ...                True                True
11        False      False  ...                False               True
12        False      False  ...                False               True
13        False      False  ...                True                True
14        False      False  ...                True                True
15        False      False  ...                False               False
16        False      False  ...                True                True
17        False      False  ...                False               False
18        False      False  ...                True                True
19        False      False  ...                True                True
20        False      False  ...                True                True
21        False      False  ...                True                True
22        False      False  ...                True                True
23        False      False  ...                True                True
24        False      False  ...                True                True
25        False      False  ...                True                True
26        False      False  ...                False               False
27        False      False  ...                True                True
28        False      False  ...                True                True
29        False      False  ...                True                True
...        ...        ...  ...                ...                ...
41417     False      False  ...                True                True
41418     False      False  ...                True                True
41419     False      False  ...                True                True
41420     False      False  ...                True                True
41421     False      False  ...                True                True
41422     False      False  ...                True                True
```

Fonte: print screen da aplicação no sistema operacional Windows 10.

A base transacionada via excel e vba gerou o seguinte resultado:

Figura 25 - Configuração utilizada no Weka para gerar as regras.



Fonte: print screen da aplicação no sistema operacional Windows 10.

Figura 26 - Associações encontradas na Base

```
Minimum support: 0.95 (39375 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 5
Size of set of large itemsets L(2): 9
Size of set of large itemsets L(3): 6
Size of set of large itemsets L(4): 1

Best rules found:

1. Vespertino=0 40937 ==> nota_parcial_cotas_alta=0 40899 <conf:(1)> lift:(1) lev:(0) [0] conv:(0.99)
2. Vespertino=0 nota_parcial_alta=0 40687 ==> nota_parcial_cotas_alta=0 40649 <conf:(1)> lift:(1) lev:(0) [0] conv:(0.98)
3. Vespertino=0 nota_parcial_cotas_media=0 39423 ==> nota_parcial_cotas_alta=0 39386 <conf:(1)> lift:(1) lev:(0) [0] conv:(0.98)
4. Integral=0 Vespertino=0 40297 ==> nota_parcial_cotas_alta=0 40259 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.97)
5. nota_parcial_alta=0 41186 ==> nota_parcial_cotas_alta=0 41147 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.97)
6. Integral=0 Vespertino=0 nota_parcial_alta=0 40080 ==> nota_parcial_cotas_alta=0 40042 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.97)
7. nota_parcial_cotas_media=0 39883 ==> nota_parcial_cotas_alta=0 39845 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.96)
8. Integral=0 40807 ==> nota_parcial_cotas_alta=0 40768 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.96)
9. nota_parcial_alta=0 nota_parcial_cotas_media=0 39697 ==> nota_parcial_cotas_alta=0 39659 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.96)
10. Integral=0 nota_parcial_alta=0 40579 ==> nota_parcial_cotas_alta=0 40540 <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.95)
```

Fonte: Print screen da aplicação no sistema operacional Windows 10.

5. RESULTADOS E DISCUSSÃO

5.1. AGRUPAMENTO

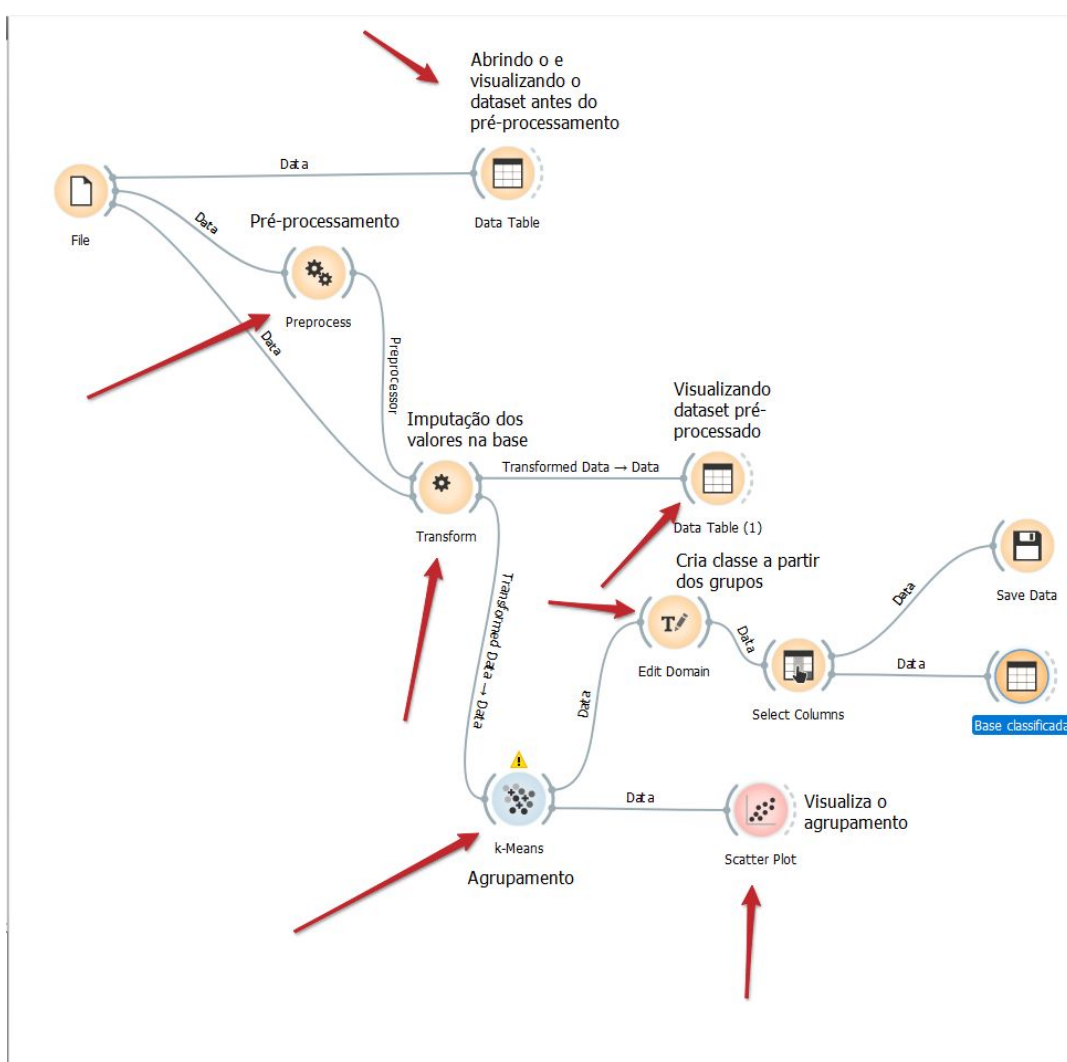
Baseando-se nos plots e informações que conseguimos retirar de nossos plots, podemos inferir que a maioria dos cursos do Grupo 3 são cursos que tendem a ser mais concorridos no *prouni*, mesmo sem os dados da concorrência do curso na base, visto a

similaridade entre os grupos podemos extrapolar para que o Grupo 2 sejam cursos com uma concorrência normal ou média, e os cursos do Grupo 1 sejam cursos de baixa concorrência. Essas conclusões se baseiam nos fatos que os cursos do Grupo 3 tem como características:

- Serem de grau bacharel
- Terem turno integral
- Oferecerem poucas bolsas

Visto isso, rotulamos a base, visando dizer se o curso tem alta, média ou baixa concorrência, para isso utilizamos a ferramenta *Orange Canvas3* que nos permitiu transformar a coluna dos grupos em uma coluna de rótulos para a classe.

Figura 27 - Workflow de classificação



Fonte: print screen da aplicação no sistema operacional Windows 10.

Após rotulada a base ficou da seguinte forma:

Figura 28 - Base rotulada

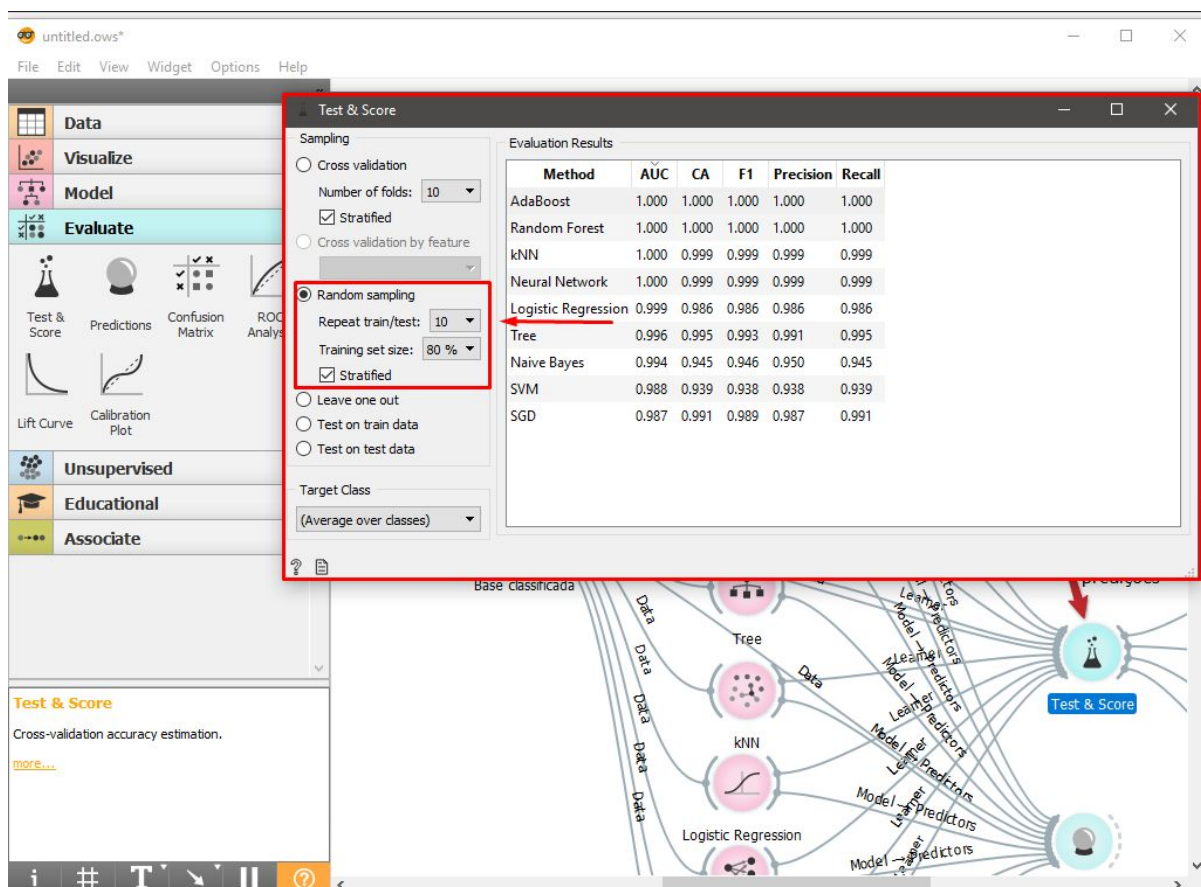
	Concorren	uf_busca	grau	turno	mensalidade	olsa_integral_cota	olsa_integral_amp	olsa_parcial_cota	olsa_parcial_amp	ota_integral_amp	ota_integral_cota
1	baixa	0.000	0.000	0.000	289.00	1.000	0.000	0.000	0.000	572.74	548.00
2	baixa	0.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	646.14	564.47
3	baixa	0.000	0.000	0.000	325.00	1.000	0.000	0.000	0.000	575.98	564.47
4	baixa	0.000	0.000	0.000	319.00	1.000	0.000	0.000	0.000	616.68	564.47
5	baixa	0.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	575.98	564.47
6	média	0.000	0.000	3.000	823.22	1.000	0.000	0.000	1.000	575.98	564.47
7	baixa	0.000	0.000	4.000	476.00	4.000	0.000	5.000	1.000	612.24	595.18
8	baixa	0.000	0.000	3.000	476.00	1.000	0.000	3.000	0.000	608.18	607.58
9	baixa	0.000	0.000	0.000	325.00	1.000	0.000	0.000	0.000	575.98	564.47
10	baixa	0.000	0.000	3.000	522.79	0.000	1.000	0.000	1.000	606.40	564.47
11	baixa	0.000	0.000	3.000	672.15	2.000	0.000	0.000	0.000	567.98	568.88
12	baixa	0.000	0.000	0.000	250.00	1.000	0.000	0.000	0.000	575.98	564.47
13	baixa	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	575.98	564.47
14	baixa	1.000	0.000	3.000	586.63	2.000	0.000	0.000	0.000	658.26	593.30
15	baixa	1.000	0.000	0.000	290.35	1.000	0.000	0.000	0.000	609.46	583.92
16	baixa	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	602.32	564.47
17	baixa	1.000	0.000	0.000	417.14	2.000	0.000	0.000	0.000	534.22	569.72
18	baixa	1.000	0.000	0.000	298.00	1.000	0.000	0.000	0.000	573.20	564.47
19	baixa	1.000	0.000	3.000	586.63	2.000	0.000	0.000	0.000	598.60	578.98
20	baixa	1.000	0.000	2.000	669.00	3.000	1.000	0.000	0.000	581.96	590.30
21	baixa	1.000	0.000	3.000	669.00	7.000	2.000	0.000	0.000	602.46	577.94
22	baixa	1.000	0.000	3.000	169.00	10.000	4.000	0.000	0.000	573.90	562.80
23	baixa	1.000	0.000	3.000	464.31	2.000	0.000	0.000	0.000	631.96	574.86
24	baixa	1.000	0.000	3.000	535.00	3.000	1.000	0.000	0.000	580.46	580.48
25	baixa	1.000	0.000	3.000	712.86	5.000	2.000	0.000	0.000	583.18	564.66
26	baixa	1.000	0.000	2.000	627.14	5.000	1.000	0.000	0.000	571.32	583.60
27	baixa	1.000	0.000	3.000	357.00	1.000	0.000	2.000	10.000	575.98	564.47
28	baixa	1.000	0.000	3.000	548.00	7.000	2.000	0.000	0.000	577.14	567.40
29	baixa	1.000	0.000	2.000	366.90	2.000	0.000	0.000	0.000	566.40	556.98
30	baixa	1.000	0.000	0.000	209.00	2.000	0.000	0.000	0.000	564.10	572.84
31	baixa	1.000	0.000	0.000	312.90	1.000	0.000	0.000	0.000	631.00	579.60
32	baixa	2.000	0.000	0.000	299.00	2.000	0.000	0.000	0.000	549.90	564.47
33	baixa	2.000	0.000	0.000	279.00	1.000	0.000	0.000	0.000	570.56	564.47
34	baixa	2.000	0.000	0.000	327.00	5.000	0.000	0.000	0.000	477.26	450.00
35	baixa	2.000	0.000	2.000	699.00	5.000	0.000	0.000	0.000	609.52	602.90
36	baixa	2.000	0.000	4.000	699.00	3.000	0.000	0.000	0.000	611.18	577.34
37	baixa	2.000	0.000	3.000	699.00	6.000	1.000	0.000	0.000	621.80	592.10
38	baixa	2.000	0.000	0.000	298.00	1.000	0.000	1.000	0.000	563.16	564.47
39	baixa	2.000	0.000	0.000	327.00	5.000	0.000	0.000	0.000	581.80	543.30
40	baixa	2.000	0.000	0.000	327.65	1.000	0.000	0.000	0.000	563.52	564.47
41	baixa	2.000	0.000	3.000	770.00	7.000	1.000	0.000	0.000	579.50	579.46
42	baixa	2.000	0.000	2.000	684.29	3.000	0.000	0.000	0.000	596.32	581.90
43	baixa	2.000	0.000	3.000	571.32	2.000	0.000	0.000	2.000	567.88	572.30
44	média	2.000	0.000	2.000	936.24	1.000	0.000	0.000	39.000	575.98	564.47
45	média	2.000	0.000	3.000	936.24	1.000	0.000	0.000	29.000	575.98	564.47
46	baixa	2.000	0.000	3.000	562.00	1.000	0.000	0.000	49.000	575.98	564.47
47	baixa	2.000	0.000	3.000	677.10	4.000	0.000	0.000	0.000	569.02	557.30
48	baixa	2.000	0.000	3.000	672.15	5.000	0.000	0.000	0.000	575.18	561.52
49	baixa	2.000	0.000	0.000	329.00	1.000	0.000	0.000	0.000	577.48	564.47
50	baixa	2.000	0.000	3.000	603.98	1.000	0.000	0.000	0.000	614.66	564.47

Fonte: print screen da aplicação no sistema operacional Windows 10.

5.2. CLASSIFICAÇÃO

No *Orange Canvas3* os algoritmos funcionam com dois nós auxiliares, o *Test and Score* e o *Predictions*, cada nó de modelo (nós rosas) efetua o treinamento da base e deve ser ligado a um *Test and Score* e/ou a um *Predictions*, o *Test and Score* irá testar a base utilizando-se de parâmetros que o usuário fornece, nesse caso testamos a base com 20% teste e 80% treinamento, o próprio *Orange* prepara a base para nós, ele “omite” as classes de 20% da base para servir como base de testes, é por isso que os modelos necessitam de um *Test and Score* para funcionar, pois eles que dão os parâmetros para o treinamento.

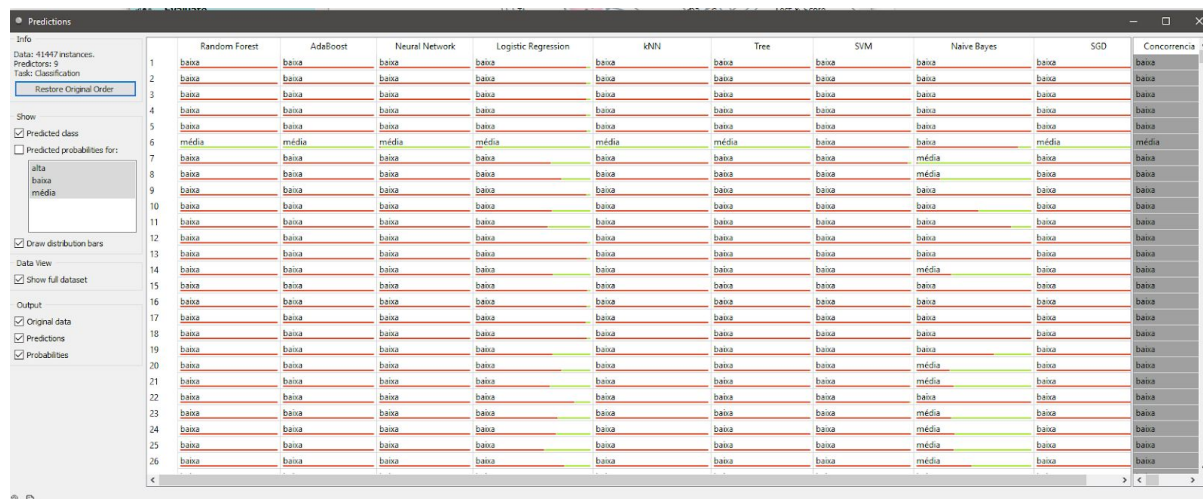
Figura 29 - Test and Score dos classificadores



Fonte: print screen da aplicação no sistema operacional Windows 10.

Observe existe uma tabela com diversas medidas para os preditores, a maioria dos preditores obtiveram uma ótima taxa de acertos, com 4 deles sendo perfeitos, atingindo 100% de acurácia. Agora para o nó *Predictions*, nós vamos ter como output uma tabela de cada modelo e suas predições para cada tupla, assim podemos comparar os vários modelos com a base original.

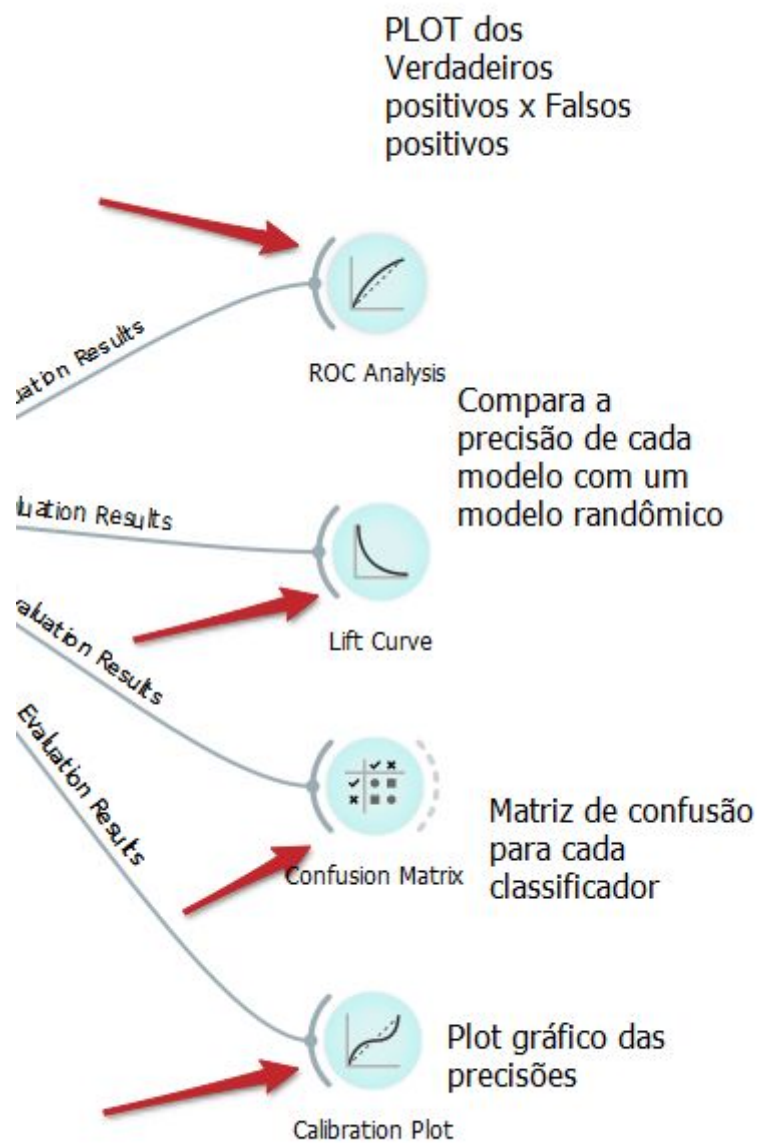
Figura 30 - Predictions dos classificadores



Fonte: print screen da aplicação no sistema operacional Windows 10.

Após a classificação nossa rotina também permite diversas visualizações sobre os preditores

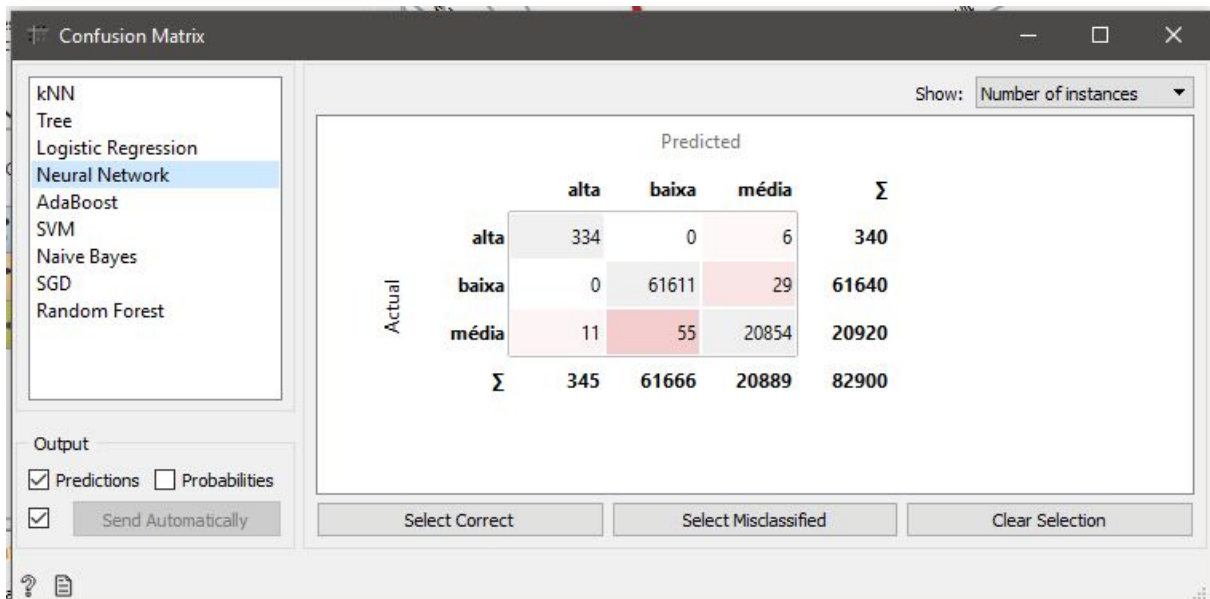
Figura 31 - Visualização de estatísticas de predições



Fonte: print screen da aplicação no sistema operacional Windows 10.

Como exemplo temos a matriz de confusão da Rede Neural que utilizamos como classificador.

Figura 32 - Matriz de confusão *Neural Network*



Fonte: print screen da aplicação no sistema operacional Windows 10.

5.3. ASSOCIAÇÃO

Baseando-se nas associações que conseguimos retirar da nossa base de dados, podemos tirar poucas conclusões relevantes a respeito da nossa base sobre notas e período do curso, mesmo tendo sido retirado os atributos como nome do curso. As regras encontradas encontram-se abaixo e em seguida todas as associações feitas.

Figura 33 - Regras de associação

```

1. Vespertino=0 40937 ==> nota_parcial_cotas_alta=0 40899 <conf:(1)> lift
2. Vespertino=0 nota_parcial_alta=0 40687 ==> nota_parcial_cotas_alta=0 4064
3. Vespertino=0 nota_parcial_cotas_media=0 39423 ==> nota_parcial_cotas_alta=
4. Integral=0 Vespertino=0 40297 ==> nota_parcial_cotas_alta=0 40259 <con
5. nota_parcial_alta=0 41186 ==> nota_parcial_cotas_alta=0 41147 <conf:(1
6. Integral=0 Vespertino=0 nota_parcial_alta=0 40080 ==> nota_parcial_cotas_
7. nota_parcial_cotas_media=0 39883 ==> nota_parcial_cotas_alta=0 39845 <
8. Integral=0 40807 ==> nota_parcial_cotas_alta=0 40768 <conf:(1)> lift:(
9. nota_parcial_alta=0 nota_parcial_cotas_media=0 39697 ==> nota_parcial_cot
10. Integral=0 nota_parcial_alta=0 40579 ==> nota_parcial_cotas_alta=0 40540

```

Fonte: print screen da aplicação no sistema operacional Windows 10.

1- Cursos no horário vespertino possuem em pelo menos 95% dos casos notas de corte até 650 pontos na modalidade parcial alta, essa regra possui confiança de 90%.

2- Cursos não integrais possuem nota de corte inferior a 650 pontos na modalidade de cotas em pelo menos 95% dos casos, e essa regra possui confiança de 92%.

A hipótese para ocorrências da primeira regra acima é de que pessoas com nota acima 650 pontos no Enem que acessam o PROUNI dão preferência para as bolsas integrais e/ou preferem o horário matutino, noturno ou integral ou até mesmo ao SISU dependendo do curso escolhido.

A hipótese para a segunda regra é de que cotistas com notas acima de 650 pontos encontram dificuldades financeiras para cursar em período integral, devido possuírem necessidades financeiras e não poderem dedicar seu tempo integralmente aos estudos. E/ou não possui dinheiro para arcar com mensalidades parcial de cursos integrais que são mais caras conforme a análise dos clusters presentes no trabalho.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BRITO, Marcelo. *Aspectos teóricos da mineração de dados e aplicação das regras de classificação para apoiar o comércio*. Disponível em: <<https://www.devmedia.com.br/aspectos-teoricos-da-mineracao-de-dados-e-aplicacao-das-regras-de-classificacao-para-apoiar-o-comercio/25429>>. Acesso em: 03 de junho de 2019.
- [2] MINERAÇÃO DE DADOS: TAREFAS E TÉCNICAS. Disponível em: <<https://www.devmedia.com.br/mineracao-de-dados-tarefas-e-tecnicas/30919>>. Acesso em 03 de junho de 2019.
- [3] CAMILO, C; SILVA, J. *Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas*. 2009. PDF.
- [4] DE CASTRO, Leandro N. *Introdução à mineração de dados*. 1ª edição. São Paulo: Saraiva, 2016.
- [5] Aggarwal, Charu C. *Data Mining: The Textbook*. Chicago: Springer, 13 de abril de 2015.
- [6] FIDELIS, Rafael. *Notas de corte do Prouni 2018 nos cursos mais procurados*. Disponível em: <<https://enemgame.com.br/blog/105-notas-de-corte-do-prouni-2018>>. Acesso em: 06 de junho de 2019.