Algoritmos de Mineração de Dados em Sistema de Monitoramento de Diabetes

Robson Cezário [1], **Alexandre de O. Zamberlan**[1]

[1]Curso de Sistemas de Informação – Universidade Franciscana (UFN)
Santa Maria – RS

{robson.cezario,alexz}@ufn.edu.br

Abstract. *This work is in the context of knowledge discovery in databases referring to diabetic patients. The database contains data on the date, time and amount of insulin used, amount of calories and carbohydrates ingested on that date, sleep quality (Likert scale) and subjective effort of physical activities (Borg scale). Therefore, this research studied, applied, evaluated and pointed out data mining methods suitable for this context. The research was based on literature review and case study, in which the work of [Rath et al. 2014] was extended. The results achieved were significant, since 3 algorithms (RandomForest, Expectation Maximisation and Bagging) returned conclusive answers.*

Resumo. *Este trabalho está no contexto de descoberta de conhecimento em bases de dados referentes a pacientes diabéticos. A base contém dados de monitoramento de data, quantidade de insulina utilizada, quantidade de calorias e carboidratos ingeridos, qualidade do sono (escala Likert) e esforço subjetivo de atividades físicas (escala Borg). Portanto, esta pesquisa buscou estudar, aplicar, avaliar e apontar métodos de mineração de dados adequados para esse contexto. A pesquisa foi baseada em revisão bibliográfica e estudo de caso, em que o trabalho de [Rath et al. 2014] foi estendido. Os resultados alcançados foram significativos, uma vez que 3 algoritmos ((RandomForest, Expectation Maximisation e Bagging)) retornaram respostas conclusivas.*

1. Introdução

A área da Saúde produz muitos e variados dados, sejam eles na Farmácia, Medicina, Enfermagem, entre outros. Tratamento de pacientes com diabetes, por exemplo, também acaba gerando uma quantia significativa e diversificada de dados, principalmente, quando pacientes, médicos, farmacêuticos e nutricionistas monitoram a evolução (ou controle) da doença. Esse controle pode ser via monitoramento de valores diários de unidades de insulina aplicada, valores diários de glicemia em jejum, quantidade de calorias e carboidratos consumidos, tempo e quantidade de exercícios realizados, entre outros. Entretanto, muitos e diferentes dados dificultam a análise e o reconhecimento de padrões que possam estar embutidos (até mesmo ocultos) numa base de dados. Assim, médicos, nutricionistas e até educadores físicos podem elaborar estratégias falhas, ou incompletas, pois a análise acaba sendo superficial.

Em 2014, um aluno do curso de Sistemas de Informação da UFN [Rath et al. 2014] projetou e implementou um Sistema de Recomendação para Diabetes, construído via a linguagem PHP e banco de dados MySQL. O projeto disponibili-

zou um sistema web para que pacientes diabéticos, seus médicos e nutricionistas pudessem registrar dados de insulina, glicemia, calorias, carboidratos, exercícios, qualidade de sono, etc. Porém, o projeto aplicou parcamente técnicas de mineração para descoberta de padrões. O projeto trabalhou com mineração, mas não era o foco principal, ou seja, o foco era o sistema web para registro e acompanhamento de dados de pessoas diabéticas. A mineração foi prototipada mais em um sentido de criar algo funcional com o mínimo possível para ser analisado e testado. Dessa forma, este trabalho estendeu a pesquisa realizada em [Rath et al. 2014], principalmente na análise e aplicação de melhores técnicas de mineração de dados na base projetada para diabetes. Registra-se que há uma base de dados criada e populada entre os anos 2012 e 2014 (independente do sistema projetado em [Rath et al. 2014]), com 722 dias monitorados. E essa base foi utilizada para o estudo, aplicação e avaliação das diferentes técnicas de mineração.

O objetivo, então, foi estudar, comparar, aplicar e avaliar algoritmos presentes no ambiente WEKA e em pacotes do universo Python para reconhecer precisamente padrões presentes na base de dados do estudo. Já os objetivos específicos foram: i) entender e aplicar as categorias de algoritmos de mineração de dados aos diferentes contextos [Tan et al. 2009], [Marques et al. 2008]; ii) estudar e avaliar algoritmos presentes no ambiente WEKA [Group 2021] e em pacotes do universo Python [Schaul et al. 2010]; iii) realizar adequação da base de dados do sistema projetado por [Rath et al. 2014]; iv) mapear e compilar trabalhos relacionados que usaram técnicas de mineração em bases de dados [Marques et al. 2008], [Vieira 2016]; v) definir, aplicar, analisar um estudo de caso para validar a proposta.

2. Revisão Bibliográfica

Nesta seção, são apresentados e discutidos conceitos que fundamentam a pesquisa e fornecem entendimento à proposta do trabalho, como descoberta de conhecimento por mineração de dados, algoritmos e ambientes de mineração, trabalhos relacionados e a doença diabetes.

2.1. Descoberta de Conhecimento e Mineração de Dados

As áreas de conhecimento produzem, diariamente, uma quantidade de dados e de informação muito grande. Esses dados e informações precisam ser coletados, tratados e armazenados, produzindo assim uma base de dados e/ou conhecimento específico.

Por exemplo, no campo da Medicina, dados coletados não são o suficiente para uma tomada de decisão precisa e eficiente. Assim, para se ter uma decisão mais assertiva, são necessárias ferramentas que facilitam e auxiliam a verificação/análise desses dados. Logo, conforme Prieto et. al. 2004 *apud* [Rath et al. 2014], as boas tomadas de decisão em relação aos quadros clínicos de pacientes diabéticos, por exemplo, influenciam diretamente no tratamento e na prevenção da doença.

Conforme Prieto et. al. 2004 *apud* [Rath et al. 2014], a análise de dados nas áreas da Saúde tem sido realizada por meio estatístico, que usa um processo matemático estabelecido com suporte teórico que permite interpretações. No entanto, há um método alternativo que auxilia em análises e interpretações de grandes quantidades de dados, conhecido como Mineração de Dados (*Data Mining* - DM). Esse método, por meio do uso ou não da estatística, busca encontrar modelos ou padrões ocultos dentro de uma base, que dificilmente seriam detectados com técnicas estatísticas.

De acordo com o trabalho de [Rath et al. 2014], alguns autores consideram os termos Mineração de Dados e Descoberta de Conhecimento em banco de dados (*Knowledge-Discovery in Databases* - KDD) como processos diferentes. De fato, mineração faz parte da linha de descoberta de conhecimento. Entretanto, neste trabalho, assume-se que os termos têm o mesmo significado: extrair conhecimento de dados.

A definição aceita e citada por vários autores sobre mineração, segundo [Rezende 2003], é: "extração de conhecimento de base de dados é o processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados".

Há variações na quantidade de etapas que compõe o processo de mineração, mas todas as etapas tratam de algumas características, como [Rezende 2003]:

- Dados: elementos armazenados em um repositório;
- Padrões: um subconjunto de dados em alguma linguagem descritiva que aponta para um tema comum;
- Processo: qualquer uma das etapas referentes à extração de conhecimento, como preparação de dados, busca por padrões e avaliação do conhecimento;
- Válidos: são os padrões encontrados obedecendo regras e/ou princípios e que sejam admissíveis;
- Novos: um padrão com um conjunto de informações definindo um novo padrão;
- Úteis: algum padrão passível de uso ou aproveitamento;
- Compreensíveis: qualquer padrão descoberto e registrado em alguma linguagem que possa ser entendida por usuários, possibilitando uma análise mais adequada dos dados;
- Conhecimento: é definido de acordo com o seu escopo de aplicação, utilidade, originalidade e compreensão, ou seja, é um conjunto de informações que fornecem algum tipo de contexto para processos de raciocínio ou inferência.

No trabalho de [Rath et al. 2014], a mineração foi dividida em 3 etapas: coleta de dados, processamento e identificação de padrões, e pós processamento.

2.2. Algoritmos clássicos para mineração de dados

De acordo com [Larose 2005], mineração de dados é classificada conforme as tarefas realizadas, sendo as mais comuns:

- Descrição: tarefa que descreve os padrões descobertos e disponibiliza uma possível interpretação dos resultados, via técnicas exploratórias de dados;
- Classificação: tem como objetivo detectar a qual classe uma determinada informação pertence, como um processo de categorização, geralmente realizada por processo de aprendizado supervisionado;
- Regressão: semelhante à classificação, mas é utilizada quando a informação é identificada por um valor numérico e não categórico;
- Predição: também semelhante à classificação e regressão, no entanto busca descobrir o valor futuro de um determinado atributo;
- Agrupamento: identifica e aproxima informações semelhantes. Um agrupamento é um conjunto de informações similares entre si, mas diferentes de outras informações nos demais agrupamentos;

- Associação: identifica a relação entre atributos de um conjunto de informações.

Segundo o trabalho realizado em [Furlan and de Souza Poletto 2018], a mineração de dados possui várias implementações distintas por meio de diversos algoritmos. Esses algoritmos são segmentados, novamente, pelas tarefas. A Figura 1 mostra uma relação dos principais algoritmos, suas descrições, tarefas e exemplos.

Técnica	Descrição	Tarefas	Exemplos
Árvore de Decisão	Baseada em estágios de decisão (nós) e na separação de classes e subconjuntos, organiza os dados de forma hierárquica.	- Classificação - Predição	CART, CHAID, C5.0, ID-3
Redes Neurais	Modelos inspirados na fisiologia do cérebro, nos quais o conhecimento é fruto do mapa de conexões neuronais e dos pesos dessas conexões.	- Classificação - Agrupamento - Predição	Perceptron, Rede MLP, Redes de Kohonen, Rede Hopfield, Rede BAM, Redes ART, Rede IAC, Rede LVQ, Rede Counterpropagation, Rede RBF, Rede PNN, Rede Time Delay, Neocognitron, Rede BSB.
Raciocínio Baseado em Casos	Baseado no método do vizinho mais próximo combina e compara atributos para estabelecer hierarquia de semelhança.	- Classificação - Agrupamento	BIRCH, CLARANS CLIQUE
Algoritmos Genéticos	Métodos gerais de busca e otimização, inspirados na Teoria da Evolução, em que a cada nova geração, soluções melhores têm mais chance de ter "descendentes".	- Classificação - Agrupamento	Algoritmo Genético Simples, Genitor, GA-Nuggets, GAPVMINER
Conjuntos Fuzzy	Oferece uma grande vantagem para classificar dados com um alto nível de abstração.	- Classificação - Agrupamento	K-means, FCMdd
Regras de Indução	Processo para obter uma hipótese a partir de dados e fatos já existentes.	- Classificação - Predição	CART, CHAID
Regras de Associação	Estabelece uma correlação estatística entre atributos de dados e conjuntos de dados.	- Associação	Apriori, AprioriTid, AprioriHybrid, AIS, SETM

Figura 1. Técnicas e Tarefas empregadas na Mineração de Dados [Goldschmidt 2005].

2.3. Ambientes ou Ferramentas para Mineração de Dados

Um dos ambientes mais conhecidos e utilizados em instituições de ensino e de pesquisa, no contexto de descoberta de conhecimento e mineração de dados, é o *Waikato Environment for Knowledge Analysis* (WEKA). É uma ferramenta de código aberto, com licença *General Public Licence* (GPL), multiplataforma (Windows, MAC OS e Linux), contendo inúmeras funções, como por exemplo, a importação de banco de dados em arquivo para mineração (*.arff*). Além disso, possui diversos algoritmos de mineração que podem ser aplicados, testados e analisados em diferentes bases importadas.

O WEKA, enquanto ferramenta, possui ambientes de importação, de visualização de resultados e de algoritmos de mineração, como algoritmos de redes neurais para aprendizado de máquina (*machine learning*), entre outros.

No ambiente de algoritmos para mineração, há sub-ferramentas de pré-processamento, oferecendo suporte para todo processo de mineração, incluindo a preparação dos dados de entrada, avaliação estatística de aprendizagem, visualização dos dados de entrada e seus resultados.

A ferramenta WEKA foi desenvolvida na linguagem JAVA, em que é possível tanto usar uma interface gráfica específica dela, quanto importar pacotes WEKA, contendo os algoritmos de mineração, em aplicações JAVA. A ferramenta possui funcionalidades divididas em ambientes, que são chamadas em botões na interface principal, como: i) *Explorer*: possui aplicações de pré-processamento, análise e visualização dos resultados; ii) *Experimenter*: aqui, o usuário pode realizar testes estatísticos entres as estruturas

de aprendizagem da ferramenta, onde se pode utilizar vários algoritmos concomitantes e comparar os resultados, escolhendo assim o algoritmo mais adequado para a base; iii) *Knowledge Flow*: com funções similares ao *Explorer*, diferencia-se pela representação gráfica dos resultados; iv) *Workbench*: ambiente que combina todas as interfaces gráficas em uma única interface; v) *Simple CLI*: ambiente para inserção de comandos em forma de *shell* ou terminal.

Existem diversos algoritmos de de mineração de dados por aprendizado de máquina implementados no WEKA (conforme já mencionado), ou seja, algoritmos que identificam padrões a partir de um conjunto de amostras de dados que servem para o treinamento prévio da base. Um dos principais destaques do ambiente WEKA nesse contexto é o *auto-weka* ou *weka.classifiers.meta.AutoWEKAClassifier* que identifica automaticamente o modelo (ou algoritmo) mais adequado com configurações de parâmetros para um determinado conjunto de dados. Especificamente, utiliza-se de várias técnicas de seleção de recursos [Kotthoff 2016].

Em geral, esse recurso funciona da seguinte maneira:

1. carregar o arquivo *.arff* com os dados a serem minerados (idêntico para qualquer outro processo no WEKA);
2. selecionar o *auto-weka*;
3. selecionar o atributo alvo ou que servirá de referência nos cruzamentos;
4. configurar parâmetros disponíveis no *auto-weka*: tamanho da memória RAM a ser utilizada no processamento e o tempo de execução desejado para o *auto-weka*;
5. iniciar o processo.

Por outro lado, o universo da linguagem Python, via bibliotecas, tem apresentado muitos recursos para os processos de descoberta de conhecimento e mineração de dados. As bibliotecas que mais tiveram destaque foram:

- Pandas: Fornece ferramentas de análise e manipulação de dados (de forma rápida e flexível) em código aberto. Pandas é adequado para diferentes tipos de dados, tais como uma tabela do padrão SQL (banco de dados relacional), planilha eletrônica, dados temporais, dados de matriz e qualquer forma de conjuntos de dados estatísticos [The pandas development team 2021];
- PyTables/HDF5: quando se tem uma quantidade muito grande de dados é utilizado o pacote Pytables (Desenvolvido com base na biblioteca HDF5) que gerencia de forma eficaz conjuntos de dados hierárquicos [PyTables Developers Team 2021];
- Theano: utilizado para determinar, incrementar e avaliar expressões matemáticas em CPUs/GPUs que envolvem matrizes multidimencionais [Laboratório LISA, Universidade de Montreal 2021];
- SeaBorn: prepara uma interface de alto nível que mostra gráficos estatísticos, interativos e informativos [Waskom 2021];
- Airflow: plataforma para criar, agendar e monitorar fluxos de trabalho [Apache Airflow 2021];
- Pybrain: é uma biblioteca de aprendizado de máquina exclusiva para Python, com uma proposta de interface amigável, com algoritmos e diferentes ambientes de teste desses algoritmos. Destaca-se, que PyBrain é uma biblioteca de aprendizagem por reforço via Redes Neurais [Schaul et al. 2010];

- Scikit-learn: possui algoritmos de classificação, regressão e agrupamento [Pedregosa et al. 2011];
- Keras: tem um interface simples para resolver problemas de aprendizado de máquina e auxilia no aproveitamento máximo de escalabilidade com redes neurais [Grupo de Interesse Especial Keras (Keras SIG) 2021];
- Dask: Utilizado para computação paralela e composto por coleções de "Big Data"e Agendamento de tarefas dinâmico [Rocklin 2015].

Assim como WEKA, o universo Python também possui uma ferramenta para identificar o modelo mais adequado para uma determinada base de dados a ser minerada. Portanto, AutoML (Auto *Machine Learning*).

2.4. Trabalhos Relacionados

Neste seção, buscou-se discutir trabalhos que abordaram ou mineração de dados e/ou mineração de dados aplicada à base de dados na área da Saúde.

O primeiro trabalho é a base para o estudo de caso desta pesquisa, onde [Rath et al. 2014] desenvolveu um sistema Web, na linguagem PHP, para cadastro e controle diário de dados do diabético, que são armazenados em banco de dados MySQL. O trabalho utilizou algoritmos presentes no ambiente WEKA de mineração.

No trabalho realizado em [Marques et al. 2008], foram aplicadas técnicas de mineração de dados para um sistema de apoio à tomada de decisão no contexto de monitoramento de jogadas em partidas de Voleibol. O sistema é conhecido como *scout*, com apelo estatístico, que possui informações muitas vezes desnecessárias, que na mineração de dados são descartadas. Dessa forma, o processo de mineração no *scout* de Voleibol foi útil para destacar informações confiáveis e algumas até desconhecidas, assim ajudando de maneira mais eficaz as comissões técnicas. O sistema foi construído na linguagem de programação Java e com o apoio do ambiente WEKA.

Segundo o que foi apresentado em [Furlan and de Souza Poletto 2018], um estudo foi realizado para aplicar conceitos de descoberta de conhecimento e de mineração em banco de dados. Alguns algoritmos do ambiente WEKA para visualização do processo foram testados.

No trabalho apresentado em [Rath et al. 2014], havia o objetivo de aplicar técnicas de mineração de dados para encontrar padrões na sua base, para poder recomendar alimentos e atividades físicas aos diabéticos. No trabalho, foi projetado e implementado um sistema Web para isso. Porém, essa recomendação não foi totalmente finalizada, uma vez que a base de dados não tinha um conjunto de dados significativo para que algoritmos de mineração fossem utilizados. Dessa forma, a principal justificativa deste trabalho. No trabalho de [Marques et al. 2008], destaca-se o processo de limpeza da base, a revisão bibliográfica das categorias dos diferentes algoritmos de mineração, facilitando a escolha de algoritmos para esta pesquisa. Já no trabalho de [Furlan and de Souza Poletto 2018], foi mostrada a análise de técnicas de algoritmos de mineração de dados presente no ambiente WEKA. Também compilou informações sobre mineração de dados, descoberta de conhecimento. Dessa forma, ficou evidente o uso da ferramenta WEKA de forma simplificada.

2.5. Monitoramento de Diabetes

De acordo com a Sociedade Brasileira de Diabetes [SBD 2021], a diabetes *mellitus* é uma doença crônica em que o corpo não é capaz de produzir ou absorver corretamente a insulina gerada pelo pâncreas. A insulina é o hormônio responsável pelo controle da quantidade de glicose no sangue que um ser humano recebe na alimentação como fonte de energia para o organismo [SBD 2021]. Novamente conforme a SBD, quando o ser humano tem diabetes e caso não for controlada de forma adequada por longos períodos, podem aparecer complicações, como doenças renais, infarto do miocárdio, acidente vascular cerebral, pé diabético, glaucoma, catarata, entre outros [SBD 2021].

Cabe ressaltar que o tratamento com insulina, busca, de forma artificial e externa ao corpo, a regulagem desse hormônio, para que cumpra seu papel no organismo. Sendo assim, portadores de diabetes Tipo 1 e Tipo 2 podem necessitar de insulina para controlar a glicose no sangue. Para essa finalidade, existem vários tipos de insulina disponíveis para o tratamento. E tal diferença se dá pelo tempo que ficam ativas no corpo, pelo tempo que levam para agir e em qual situação do dia são mais eficientes.

Independente do tipo de diabetes ou o tipo de terapia ou tratamento, o monitoramento é essencial para o controle, para o tratamento, para o acompanhamento dos profissionais Médicos, Nutricionistas e Educadores Físicos. Há profissionais da área da Saúde que monitoram seus pacientes com exames de sangue tradicionais, realizados de tempos em tempos, como glicemia em jejum, glicemia glicada, insulina, triglicerídeos, colesterol, etc. Porém, há outros tipos de acompanhamento, ditos diários: glicemia, unidades aplicadas de insulina, qualidade do sono, quantidade de calorias e carboidratos ingeridos, tempo e tipo de atividade física.

3. Proposta de trabalho

A partir do exposto na seção anterior, esta pesquisa estendeu o trabalho [Rath et al. 2014] e aplicou algoritmos de mineração em base com um conjunto significativo de dados para que os algoritmos possam retornar resultados reais e mais fidedignos.

3.1. Materiais e métodos

Este trabalho foi baseado em pesquisa exploratória com revisão bibliográfica amparado com estudo de caso. O estudo de caso foi a aplicação de mineração na base de dados criada entre 2012 e 2014, modelagem e prototipação de sistema Web para gestão de dados de pacientes diabéticos. Já no projeto e desenvolvimento da solução, foram utilizados a metodologia *Scrum* [Wykowski and Wykowska 2019] com a técnica *Kanban*.

As ferramentas utilizadas foram: Trello - técnica kanban; GitHub - controle de versão; Astah - diagramação UML; Ambiente WEKA - simulador para testes de algoritmos de mineração de dados; Pacotes do universo Python para mineração em banco de dados.

3.2. Modelagem do sistema

Em termos funcionais da proposta do estudo de caso, que também contempla um sistema Web adaptado do trabalho de [Rath et al. 2014], na Figura 2 é possível entender todos os atores que podem ter relação com o sistema, suas principais funcionalidades e a funcionalidade de aplicação de algoritmos de descoberta de conhecimento.

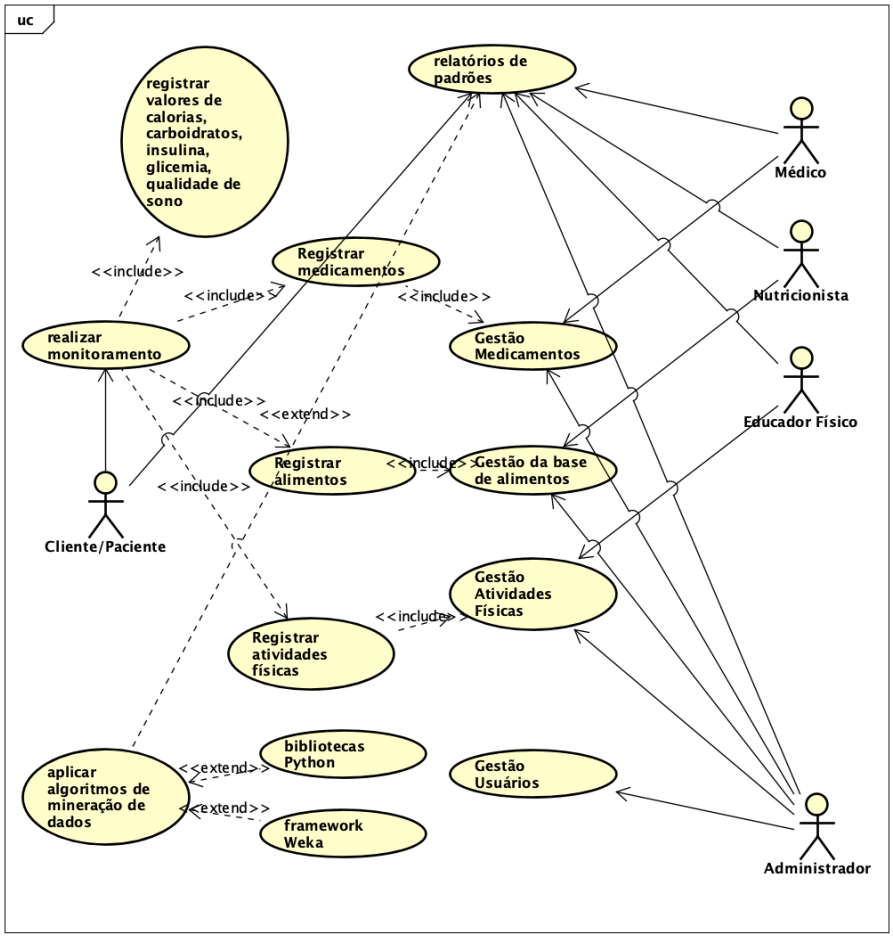


Figura 2. Diagrama de Casos de Uso para o estudo de caso adaptado do sistema de [Rath et al. 2014].

Já em relação à organização do protótipo do estudo de caso, pensou-se orientado a objetos tendo como referência o padrão arquitetural *Model-View-Template* (MVT) utilizado pelo *framework* Django. Na Figura 3, é possível visualizar os pacotes e as classes que fazem a persistência no banco de dados na camada *model*.

Registra-se que o *framework* Django utiliza de forma automática o Mapeamento Objeto-Relacional (MOR), ou seja, a geração do banco de dados é realizada a partir das classes existentes na camada *model* do sistema. Assim, a Figura 3 mostra a estrutura de pacotes/classes e, de forma equivalente, o modelo Entidade-Relacionamento.

3.3. Preparando a base para a mineração

A Figura 4 mostra como os dados monitorados (dias da semana, ano, glicemia, quantidade de insulina, quantidade calorias, quantidade de carboidratos, qualidade do sono, tempo de atividades físicas) foram coletados. A ressalva é que a média de glicemia era analisada sempre de 3 em 3 dias, pois essa média sugeria alteração nas unidades de insulina aplicada. Ou seja, se a média fosse maior que 100 mg/dL, a quantidade de unidades de insulina deveria ser aumentada em duas. Caso fosse menor que 80 mg/dL, deveria diminuir em 2 unidades.

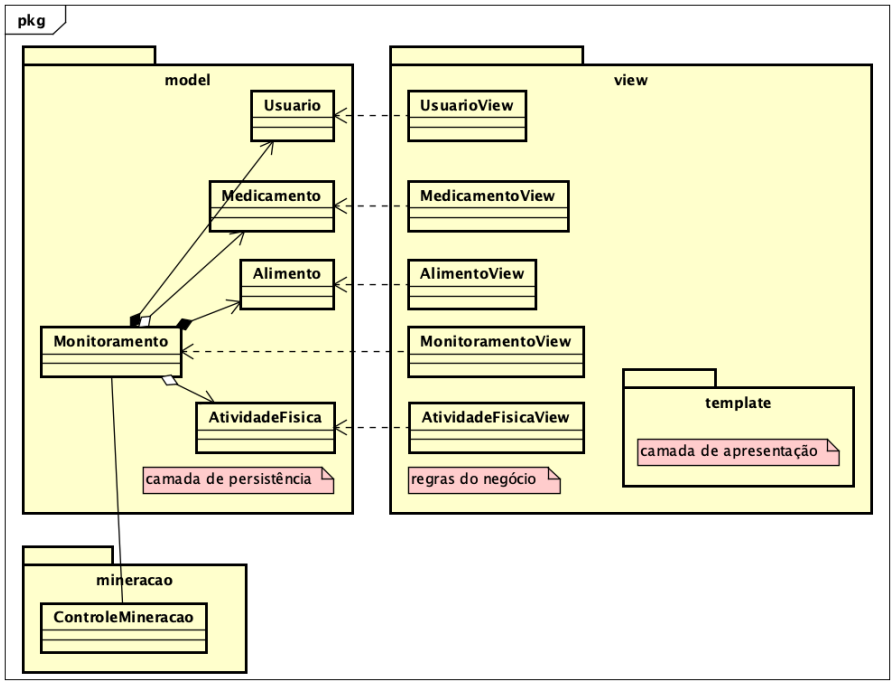


Figura 3. Diagrama de pacotes ou de domínio do protótipo.

Dia Semana	Data	Antes Comer / Depois Comer	Resultado Glicemia	Dose Insulina	kcal	carb	noite de sono 1-5	padel	musculacao R	musculacao H	pilates	corrida	caminhada	tenis	sauna	bike	natacao	eliptico	volei de areia
Quinta	2012	ac	Normal	6	Abaixo	Acima	4	0	0	0	0	0	0	0	0	0	0	0	0
Sexta	2012	ac	Normal	6	Recomendado	Acima	4	0	13	0	0	11	0	0	0	0	0	0	0
Sabado	2012	ac	Normal	6	Recomendado	Acima	4	17	0	0	0	0	0	0	0	0	0	0	0
Domingo	2012	ac	Acima	6	Abaixo	Acima	5	0	0	0	0	0	0	0	0	0	0	0	0
Segunda	2012	ac	Normal	6	Recomendado	Acima	5	17	0	0	0	0	0	0	0	0	0	0	0
Terca	2012	ac	Normal	6	Recomendado	Acima	4	0	0	0	0	0	0	0	0	0	0	0	0
Quarta	2012	ac	Normal	6	Recomendado	Acima	5	0	0	0	0	0	0	13	0	0	0	0	0
Quinta	2012	ac	Abaixo	6	Recomendado	Acima	4	0	0	0	0	0	0	0	0	0	0	0	0
Sexta	2012	ac	Abaixo	6	Recomendado	Acima	5	0	0	0	0	0	0	0	0	0	0	0	0
Sabado	2012	ac	Normal	4	Recomendado	Recomendado	3	17	0	0	0	0	0	0	0	0	0	0	0
Domingo	2012	ac	Normal	4	Recomendado	Recomendado	5	0	0	0	0	0	0	0	0	0	0	0	0
Segunda	2012	ac	Normal	4	Recomendado	Acima	4	0	13	0	0	15	0	0	0	0	0	0	0
Terca	2012	ac	Normal	4	Recomendado	Acima	5	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4. Planilha eletrônica com melhorias nas colunas e nos dados, para facilitar a conversão ao arquivo *.arff* (arquivo de mineração no WEKA).

Destaca-se que a plataforma WEKA precisa receber uma base tratada, preferencialmente com valores categóricos, assim foi preciso melhorar alguns campos (colunas ou atributos). Dessa forma, foram realizadas conversões de valores numéricos para valores categóricos, principalmente nos campos medição de glicemia, quantidade de exercícios realizados, quantidade de carboidratos, quantidade de calorias ingeridos em um dia e resultado da medição da glicemia.

Na medição de glicemia, utilizou-se abaixo, normal (ou recomendado) e acima. Porém, há dois momentos de medição: em jejum e pós-prandial (após 2 horas de uma refeição). Portanto, para medições em jejum, o abaixo do recomendado é um valor menor que 80 mg/dL, recomendado ou normal é um valor entre 80 e 100 mg/dL e acima do recomendado um valor maior que 100 mg/dL. Para medições pós-prandial, o normal é um valor abaixo de 140 mg/dL e ”acima”, um valor maior que 140 mg/dL.

Em relação à quantidade de exercícios, optou-se em utilizar a escala (ou tabela) de

Borg (de 0 a 20), que trabalha com uma percepção subjetiva de quanto esforço o próprio atleta realizou, sendo que 1 é quase nenhum esforço e 20 o esforço máximo possível [Borg and Borg 2001]. De acordo com [Borg and Borg 2001], os valores entre 6 e 20 são baseados na Frequência Cardíaca de 60 a 200 batimentos por minuto (bpms), sendo que o valor 12 corresponde aproximadamente 55% e o 16 a 85% da Frequência Cardíaca Máxima. Dessa forma, a escala corresponderia ao esforço da seguinte forma: 7 - muito fácil; 9 - fácil; 11 - relativamente fácil; 13 - ligeiramente cansativo; 15 - cansativo; 17 - muito cansativo; 19 - exaustivo.

Em relação ao consumo de carboidratos e calorias, também foram necessárias modificações na tabela original, transformando dados numéricos em dados categóricos, como abaixo, recomendado e acima do recomendado. Dessa forma, se uma pessoa tem como orientação (referência), na sua dieta, ingerir no máximo 200 gramas carboidratos e 2600 calorias, o abaixo do recomendado é quando essa pessoa ingere menos de 80% desses valores de referência, recomendado quando ingere entre 80% a 120% dos valores e acima do recomendado quando consome mais de 120% dos valores sugeridos na dieta. Essas orientações são estratégias básicas no contexto da nutrição.

```
1  @relation 'Registros de Glicose Alexandre - paraArtigo'
2  @attribute 'Dia Semana' {Quinta,Sexta,Sabado,Domingo,Segunda,Terca,Quarta}
3  @attribute Data {2012,2013,2014}
4  @attribute 'Antes Comer / Depois Comer ' {ac,dc}
5  @attribute 'Resultado Glicemia' {Normal,Acima,Abaixo}
6  @attribute 'Dose Insulina' numeric
7  @attribute kcal {Abaixo,Recomendado,Acima}
8  @attribute carb {Acima,Recomendado,Abaixo}
9  @attribute 'noite de sono' {1,2,3,4,5}
10 @attribute padel {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
11 @attribute 'musculacao R' {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
12 @attribute 'musculacao H' {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
13 @attribute pilates {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
14 @attribute corrida {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
15 @attribute caminhada {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
16 @attribute tenis {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
17 @attribute sauna {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
18 @attribute bike {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
19 @attribute natacao {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
20 @attribute eliptico {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
21 @attribute 'volei de areia' {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
22 @data
23 Quinta,2012,ac,Normal,6,Abaixo,Acima,4,0,0,0,0,0,0,0,0,0,0,0,0
24 Sexta,2012,ac,Normal,6,Recomendado,Acima,4,0,13,0,0,11,0,0,0,0,0,0,0
25 Sabado,2012,ac,Normal,6,Recomendado,Acima,4,17,0,0,0,0,0,0,0,0,0,0,0
26 Domingo,2012,ac,Acima,6,Abaixo,Acima,5,0,0,0,0,0,0,0,0,0,0,0,0
27 Segunda,2012,ac,Normal,6,Recomendado,Acima,5,17,0,0,0,0,0,0,0,0,0,0,0
28 Terca,2012,ac,Normal,6,Recomendado,Acima,4,0,0,0,0,0,0,0,0,0,0,0,0
29 Quarta,2012,ac,Normal,6,Recomendado,Acima,5,0,0,0,0,0,0,13,0,0,0,0,0
30 Quinta,2012,ac,Abaixo,6,Recomendado,Acima,4,0,0,0,0,0,0,0,0,0,0,0,0
```

Figura 5. Dados tratador pela plataforma WEKA e armazenados em arquivo *.arff*.

Uma vez melhorada a tabela, já é possível no ambiente WEKA importar a base da planilha eletrônica. A importação (automática) converte a tabela em um arquivo com extensão *.arff*, com os campos (colunas) para sua manipulação interna, conforme mostra a Figura 5.

Ressalta-se que *@attribute* representa cada coluna da planilha eletrônica. Alguns atributos precisam ser valorados ou categóricos, como por exemplo, os atributos *Dia Semana* que contém os dias da semana, *Resultado Glicemia* que contém as categorias Normal, Acima e Abaixo. Finalmente, a partir da linha 23, há os dados da planilha convertidos para a plataforma WEKA. Por exemplo, na linha 23 há uma medição referente a

uma quinta-feira, do ano de 2012, com uma medição antes de comer, com resultado de glicemia Normal, 6 unidades de insulina aplicada, calorias Abaixo do recomendado, carboidratos Acima do recomendado, 4 pontos na qualidade do sono e nenhuma atividade física realizada.

3.4. Aplicando os algoritmos de mineração via WEKA

Uma vez que a base foi preparada para ser trabalhada no WEKA, testaram-se todos as categorias de algoritmos presentes no ambiente que ficaram disponíveis para a base cadastrada. Ou seja, alguns algoritmos não foram habilitados à execução dessa base. Independente dos algoritmos que ficaram disponíveis para esse formato de base de dados, o ambiente WEKA permite selecionar atributos para serem trabalhados, como ilustra a Figura 6, em que alguns dados foram selecionados.

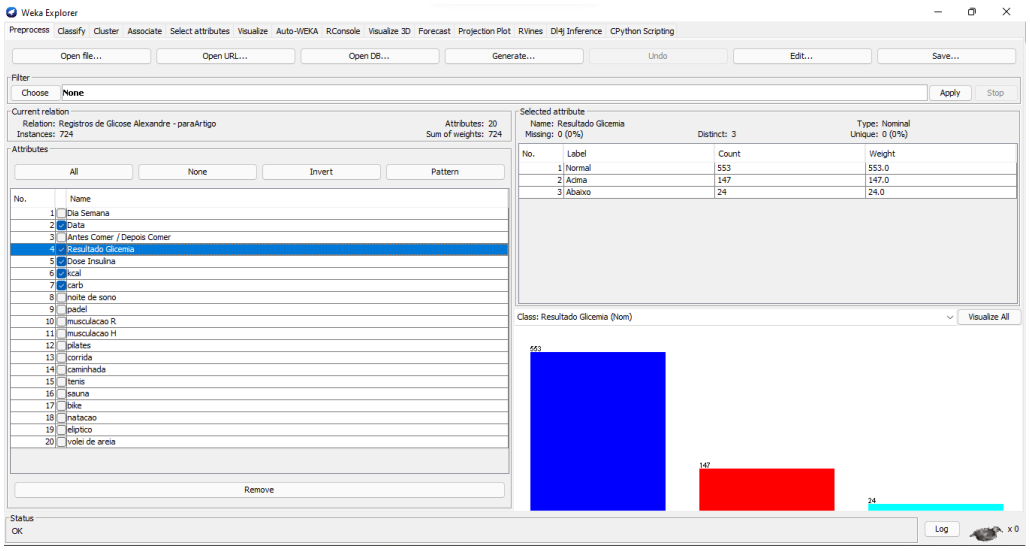


Figura 6. WEKA na aba preprocess com atributos selecionados.

Dos algoritmos de mineração de dados (mencionados na Revisão Bibliográfica), o algoritmo *RandomForest* (Figuras 7), da categoria de classificação em árvores de decisão, apresentou resultados com quase 97% de assertividade, além de trazer uma matriz de confusão que identificou (Figura 8) 550 dias de glicemia normal, também como mostrado na Figura 6, validando o resultado.

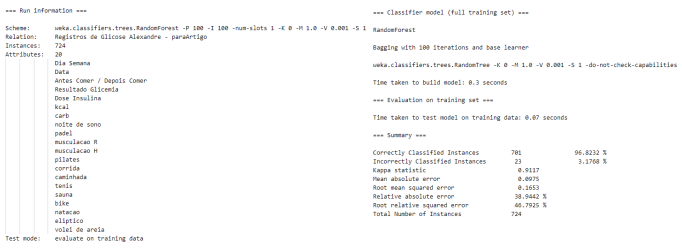
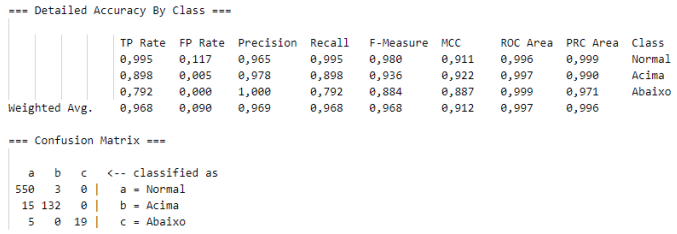


Figura 7. Aplicação do algoritmo *RandomForest*.

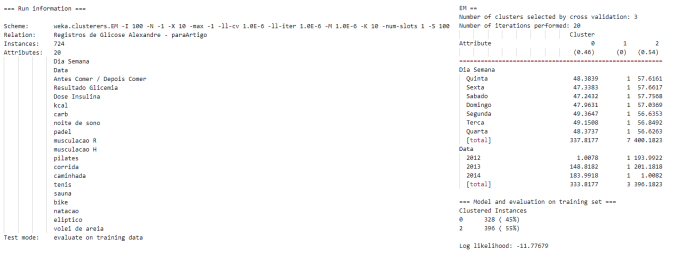
Em relação aos algoritmos de agrupamento (clusterização), *Expectation Maximisation* (EM) foi o que facilitou a análise e conclusões dos resultados (fácil interpretação).



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,995    0,117    0,965      0,995    0,980      0,911    0,996     0,999     Normal
               0,898    0,005    0,978      0,898    0,936      0,922    0,997     0,990     Acima
               0,792    0,000    1,000      0,792    0,884      0,887    0,999     0,971     Abaixo
Weighted Avg.  0,968    0,090    0,969      0,968    0,968      0,912    0,997     0,996

=== Confusion Matrix ===

   a   b   c   <-- classified as
 550   3   0 |   a = Normal
  15 132   0 |   b = Acima
   5   0  19 |   c = Abaixo
```

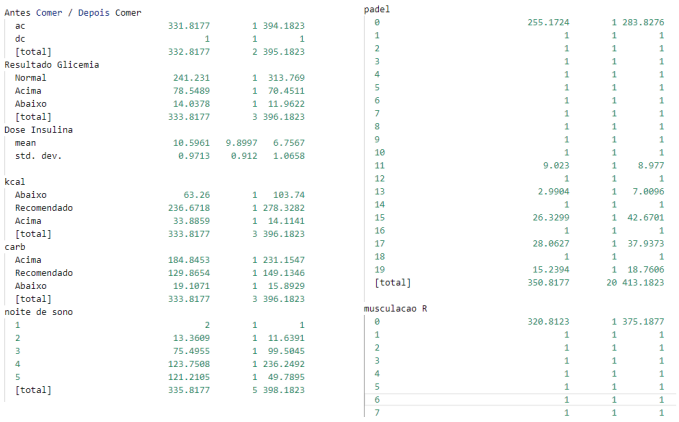
Figura 8. Aplicação do algoritmo *RandomForest* com a matriz de confusão.

Os resultados parciais obtidos pelo EM podem ser visualizados nas Figuras 9 e 10. Note, que há 3 colunas representando os grupos criados ou identificados.



```
=== Run information ===

Scheme:       weka.clusterers.EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100
Relation:     Registros da Glicose Alexandre - paraArtigo
Instances:    724
Attributes:   20
              Dia Semana
              Data
              Antes Comer / Depois Comer
              Resultado Glicemia
              Dose Insulina
              kcal
              carb
              noite de sono
              padel
              musculacao R
              musculacao H
              pilates
              corrida
              caminhada
              tenis
              sauna
              bike
              natacao
              eliptico
              volei de areia
Test mode:    evaluate on training data

EM ==
Number of clusters selected by cross validation: 3
Number of iterations performed: 20
                                Cluster
Attribute                            0          1          2
                                  (0.46)       (0)    (0.54)
==========================================================================
Dia Semana
  Quinta                         48.3839          1    57.6161
  Sexta                          47.3383          1    57.6617
  Sabado                         47.2432          1    57.7568
  Domingo                        47.9631          1    57.0369
  Segunda                        49.3647          1    56.6353
  Terca                          49.1508          1    56.8492
  Quarta                         48.3737          1    56.6263
  [total]                       337.8177          7  400.1823
Data
  2012                            1.0078          1  193.9922
  2013                          148.8182          1  201.1818
  2014                          183.9918          1    1.0082
  [total]                       333.8177          3  396.1823

=== Model and evaluation on training set ===
Clustered Instances
0      328 ( 45%)
2      396 ( 55%)

Log likelihood: -11.77679
```

Figura 9. Aplicação do algoritmo EM e o agrupamento da base em 3 grupos.



```
Antes Comer / Depois Comer
  ac                          331.8177          1  394.1823
  dc                                 1          1         1
  [total]                    332.8177          2  395.1823
Resultado Glicemia
  Normal                      241.231          1   313.769
  Acima                       78.5489          1   70.4511
  Abaixo                      14.0378          1   11.9622
  [total]                    333.8177          3  396.1823
Dose Insulina
  mean                        10.5961     9.8997    6.7567
  std. dev.                    0.9713      0.912    1.0658

kcal
  Abaixo                        63.26          1    103.74
  Recomendado                236.6718          1  278.3282
  Acima                        33.8859          1   14.1141
  [total]                    333.8177          3  396.1823
carb
  Acima                      184.8453          1  231.1547
  Recomendado                129.8654          1  149.1346
  Abaixo                      19.1071          1   15.8929
  [total]                    333.8177          3  396.1823
noite de sono
  1                                  2          1         1
  2                           13.3609          1   11.6391
  3                           75.4955          1   99.5045
  4                          123.7508          1  236.2492
  5                          121.2105          1   49.7895
  [total]                    335.8177          5  398.1823

padel
  0                          255.1724          1  283.8276
  1                                 1          1         1
  2                                 1          1         1
  3                                 1          1         1
  4                                 1          1         1
  5                                 1          1         1
  6                                 1          1         1
  7                                 1          1         1
  8                                 1          1         1
  9                                 1          1         1
  10                                1          1         1
  11                            9.023          1     8.977
  12                                1          1         1
  13                           2.9904          1    7.0096
  14                                1          1         1
  15                          26.3299          1   42.6701
  16                                1          1         1
  17                          28.0627          1   37.9373
  18                                1          1         1
  19                          15.2394          1   18.7606
  [total]                    350.8177         20  413.1823

musculacao R
  0                          320.8123          1  375.1877
  1                                 1          1         1
  2                                 1          1         1
  3                                 1          1         1
  4                                 1          1         1
  5                                 1          1         1
  6                                 1          1         1
  7                                 1          1         1
```

Figura 10. Aplicação do algoritmo EM e o agrupamento da base em 3 grupos: dados insulina, glicemia, carboidrato, calorias, atividades físicas.

E com isso, foi possível concluir que:

- foram identificados 314 dias com controle de glicemia (agrupamento 2), dessa forma, esse agrupamento foi a referência para as conclusões;
- exercícios com carga muscular elevada ajudam mais no controle da glicemia, logo na aplicação diminuída de insulina (média de 7 unidades com desvio padrão de 1.0);
- a variável sono tem influência na quantidade de glicemia, uma vez que no grupo 2 apresentou valores significativos de qualidade 4 e 5;
- a quantidade elevada consumida de carboidratos no grupo 2 (que é o grupo com glicemia controlada), indica que a prática de exercícios foi decisiva.

Além desses algoritmos, aplicou-se o *auto-weka* com memória configurada para 1 Gbytes nos tempos de 15, 60, 120 e 240 minutos, enquanto que o último teste foi com memória configurada em 5 Gbytes e com tempo 1440 minutos. Registra-se que independente da quantidade de memória ou de tempo, todas os testes apresentaram resultados similares. A Figura 11 mostra os atributos, o total de instâncias, as configurações do último teste (5 Gb e 1440 minutos de processamento total). E o algoritmo ou modelo sugerido pelo *auto-weka* foi o Bagging.

```
1  === Run information ===
2  Scheme:       weka.classifiers.meta.AutoWEKAClassifier -seed 123 -timeLimit 1440 -memLimit 5120 -nBestConfigs 1 -metric errorRate
   -parallelRuns 1
3  Relation:     Registros de Glicose Alexandre - paraArtigo
4  Instances:    724
5  Attributes:   20
6                Dia Semana
7                Data
8                Antes Comer / Depois Comer
9                Resultado Glicemia
10               Dose Insulina
11               kcal
12               carb
13               noite de sono
14               padel
15               musculacao R
16               musculacao H
17               pilates
18               corrida
19               caminhada
20               tenis
21               sauna
22               bike
23               natacao
24               eliptico
25               volei de areia
26 Test mode:    evaluate on training data
```

Figura 11. Relatório da aplicação do auto-weka na base, tendo como atributo alvo o resultado da glicemia.

```
28 === Classifier model (full training set) ===
29 best classifier: weka.classifiers.meta.Bagging
30 arguments: [-P, 41, -I, 12, -S, 1, -W, weka.classifiers.bayes.NaiveBayes, --, -K]
31 attribute search: weka.attributeSelection.GreedyStepwise
32 attribute search arguments: [-C, -B, -N, 16]
33 attribute evaluation: weka.attributeSelection.CfsSubsetEval
34 attribute evaluation arguments: [-M, -L]
35 metric: errorRate
36 estimated errorRate: 0.23342541436464087
37 training time on evaluation dataset: 0.009 seconds
38
39 Correctly Classified Instances         555               76.6575 %
40 Incorrectly Classified Instances       169               23.3425 %
41 Kappa statistic                        0.0181
42 Mean absolute error                    0.1691
43 Root mean squared error                0.3637
44 Relative absolute error                67.5416 %
45 Root relative squared error            102.9767 %
46 Total Number of Instances              724
47
48 === Confusion Matrix ===
49
50    a   b   c   <-- classified as
51  553   0   0 |   a = Normal
52  145   2   0 |   b = Acima
53   24   0   0 |   c = Abaixo
```

Figura 12. Relatório final mostrando a escolha do melhor método de mineração *weka.classifiers.meta.Bagging*.

Na Figura 12, os resultados da mineração pelo algoritmo *weka.classifiers.meta.Bagging* colaboraram com os resultados obtidos pelo algoritmo *RandomForest*. Perceba nessa figura que, a partir da matriz de confusão gerada, 553 dias foram glicemia normal (enquanto no *RandomForest* foram 550 dias).

O próprio *auto-weka*, ao finalizar seu processo, recomendou para um melhor desempenho dar mais tempo ao processamento, pois há 348 configurações possíveis para obter bons resultados de forma confiável.

3.5. SOMDiabetes - Sistema Online de Monitoramento da Diabetes

O protótipo do estudo de caso, já possui um conjunto de funcionalidades básicas de gestão de usuários (pacientes, nutricionistas, educadores físicos, médicos), de alimentos e de atividades físicas. Além disso, há toda a gestão de registro de refeições por paciente em

um determinado dia, com contagem de calorias e carboidratos. Também, há a gestão de registro de atividades físicas realizadas em um período. Dessa forma, as Figuras 13, 14, 15 e 16 mostram essas funcionalidades em operação.

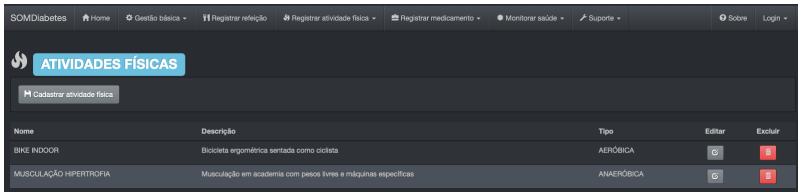


Figura 13. Interface gestão de atividades físicas.

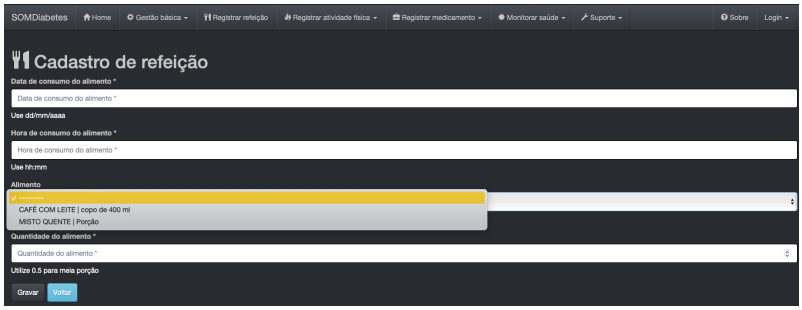


Figura 14. Interface de registro de refeições em um dia e hora.

Figura 15. Interface da lista de refeições registradas.

Figura 16. Interface da lista de atividades registradas.

4. Conclusões

Ao longo do texto, foram apresentados e discutidos conceitos referentes a diabetes (como uma doença que exige acompanhamento para um tratamento mais preciso), descoberta de conhecimento e a reconhecimento de padrões por meio de mineração de dados. Também foram analisados trabalhos relacionados que forneceram suporte a esta pesquisa, principalmente, o trabalho de Rodrigo Rath [Rath et al. 2014].

Os resultados atingidos foram a remodelagem de alguns aspectos estruturais e funcionais existentes em [Rath et al. 2014], a adequação da planilha eletrônica (com dados de monitoramento de 3 anos de um doente em diabetes) ao formato do ambiente WEKA (com transformação de dados numéricos para dados categóricos). Além disso, a identificação de 3 modelos que melhor atenderam a base de testes (RandomForest, EM e Bagging), pois classificaram e/ou aglomeraram a base tendo como atributo alvo o resultado da glicemia. Dessa forma, os demais atributos foram cruzados com o resultado da glicemia.

Finalmente, registram-se alguns trabalhos futuros:

- converter e aplicar os algoritmos do WEKA em algoritmos do universo Python, com o *framework* Pandas, que trata do banco de dados (*dataset*), e do pacote Scikit-learn;
- importar os dados da planilha melhorada do WEKA para o banco de dados do sistema SOMDiabetes;
- finalizar os cálculos de gastos calóricos quando atividades físicas realizadas;
- importar a lista de alimentos da Sociedade Brasileira de Nutrição para o banco de dados do sistema SOMDiabetes;
- criar as funcionalidades para que nutricionista, educador físico e médico possam acompanhar seus pacientes dentro do sistema SOMDiabetes;

Referências

Apache Airflow (2021). Apache Airflow. http://airflow.apache.org/. Acessado em Março de 2021.

Borg, G. and Borg, E. (2001). A new generation of scaling methods: Level-anchored ratio scaling. *Psychologica*, 28:15–45.

Furlan, M. B. and de Souza Poletto, A. S. R. (2018). Algoritmos e técnicas para mineração de dados. *Fundação Educacional do Município de Assis/SP*.

Goldschmidt, E. P. . R. (2005). *Data Mining: Um Guia Prático*. Editora Elsevier.

Group, M. L. (2021). Weka: The workbench for machine learning. https://www.cs.waikato.ac.nz/ml/weka/. Acessado em Março de 2021.

Grupo de Interesse Especial Keras (Keras SIG) (2021). Keras SIG. https://keras.io/. Acessado em Maio de 2021.

Kotthoff, L. (2016). Auto Weka. https://www.cs.ubc.ca/labs/beta/Projects/autoweka/. Acessado em Novembro de 2021.

Laboratório LISA, Universidade de Montreal (2021). Biblioteca Python: Teano. https://pypi.org/project/Theano//. Acessado em Maio de 2021.

Larose, D. T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. Editora John Wiley and Sons.

Marques, É. B., Zamberlam, A. d. O., de Oliveira, R. F., Raimann, L. H., and de Oliveira, L. V. (2008). Projeto de módulo de data mining para scout voleibol. *Seminário de Informática-RS (SEMINFO RS 2008), Torres (RS)*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

PyTables Developers Team (2021). PyTables: Hierarchical Datasets in Python. http://www.pytables.org/. Acessado em Maio de 2021.

Rath, R., Zamberlan, A., and Vieira, S. (2014). Sistema de recomendação para controle da diabetes. In *7o Congresso Sul Brasileiro de Computação*, Criciúma. SULCOMP, UNESC.

Rezende, S. O. (2003). *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda.

Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. In Huff, K. and Bergstra, J., editors, *Proceedings of the 14th Python in Science Conference*, pages 130 – 136.

SBD (2021). Sociedade Brasileira de Diabetes. https://www.diabetes.org.br/. Acessado em Março de 2021.

Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Rückstieß, T., and Schmidhuber, J. (2010). PyBrain. *Journal of Machine Learning Research*, 11:743–746.

Tan, P.-N., Steinbach, M., and Kumar, V. (2009). *Introdução ao datamining: mineração de dados*. Ciência Moderna, São Paulo.

The pandas development team (2021). pandas-dev/pandas: Pandas. https://pandas.pydata.org/. Acessado em Maio de 2021.

Vieira, S. A. G. (2016). *Identificaćão de padrões de expressão em doenćas genéticas usando uma rede de integraćão de vias de manutenćão do Genoma, Angiogênese, Hipóxia e Vigilância Imunológica*. PhD thesis, Centro Universitário Franciscano, Santa Maria.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wykowski, T. and Wykowska, J. (2019). Lessons learned: Using scrum in non-technical teams. https://www.agilealliance.org/resources/experience-reports/lessons-learned-using-scrum-in-non-technical-teams/. Acessado em Março de 2021.