

Trabalho Final - Ciência de Dados

Comparação de Classificadores

João Vitor Barbosa Dantas – 12311BSI214
Matheus Gualter Silva Resende – 12311BSI310
Pedro Emilio Castro Lemos – 12311BSI222

Introdução e Objetivos

A gestão dos Restaurantes Universitários (RUs) é um desafio complexo que impacta diretamente a vida dos estudantes. Historicamente, a gestão dos RUs da Universidade Federal de Uberlândia (UFU) dependeu de metodologias com baixa adesão para o monitoramento da qualidade, além de possuir um histórico de consumo e planejamento de cardápios inacessível, frequentemente armazenado em formatos não estruturados. Diante desse cenário, a aplicação de técnicas de Ciência e Engenharia de Dados torna-se fundamental para estruturar essas informações e extrair inteligência para a tomada de decisão.

Neste contexto, este trabalho tem como premissa a utilização de um conjunto de dados inédito para comunidade acadêmica geral da UFU, disponível em: <https://github.com/MatheuxGualter/classifiers.git>, consolidado a partir da rotina operacional e da percepção da comunidade acadêmica sobre as refeições servidas. A partir dessa base estruturada, exploramos a aplicação de algoritmos de aprendizado de máquina supervisionado para resolver dois problemas centrais da administração do RU: a previsão do volume de demanda diária e a classificação do nível de satisfação dos usuários em relação aos pratos oferecidos no cardápio.

Para alcançar esse propósito, o objetivo central deste relatório consiste em aplicar, avaliar e interpretar o desempenho de diferentes algoritmos de classificação sobre essas duas bases de dados distintas. De forma específica, o trabalho visa conduzir um pré-processamento detalhado dos dados brutos, empregando técnicas de engenharia de atributos para tratar desbalanceamentos e ruídos de amostragem, justificar a escolha dos classificadores utilizados para cada cenário e, por fim, realizar uma avaliação crítica das métricas de resultados obtidas, consolidando os conhecimentos teóricos e práticos da disciplina.

Descrição das bases

As bases de dados utilizadas neste trabalho são de autoria própria, originadas e estruturadas durante o projeto de pesquisa "Ciência de dados aplicada à satisfação do usuário nos Restaurantes Universitários" (Projeto DataRU), previamente validadas pela professora da disciplina, cumprindo o requisito de escolha da base.

1. [Base de Dados de Consumo \(Previsão de Demanda\)](#): Esta base compila os dados quantitativos diários de acesso aos refeitórios.

DATA	TIPO...	...BOLSISTAS	...FUNCIONÁRIOS	...ALUNOS	TOTAL	DIA...	DEMANDA
2024-01-08	almoço	111	0	858	969	1	NORMAL

2024-01-08	jantar	145	0	1478	1623	1	NORMAL
2024-01-10	jantar	163	0	1695	1858	3	ALTA
2024-07-15	almoço	71	0	445	516	1	BAIXA
2024-07-18	jantar	91	3	558	652	4	BAIXA

- **Atributos e Tipos de Dados:** A base é composta por atributos numéricos contínuos (quantidades e valores financeiros de acessos de bolsistas, alunos e funcionários, além dos totais consolidados como QNT_TOTAL e VAL_TOTAL) e atributos categóricos/temporais (DATA, TIPO_REFEICAO e DIA_SEMANA).
- **Valores Ausentes:** A base original, não estruturada não possuía valores ausentes, assim, não foi necessário tratá-los.
- **Atributo Classe:** O atributo alvo criado para a classificação é a DEMANDA, uma variável categórica segmentada em três valores: ALTA, NORMAL e BAIXA.
- **Distribuição e Balanceamento:** Trata-se de um problema propositalmente desbalanceado de forma leve. A discretização da classe foi feita através de percentis relativos ao dia da semana e tipo de refeição, alocando os 25% dias de maior movimento na classe ALTA, os 25% de menor movimento na classe BAIXA, e os 50% centrais na classe NORMAL. Essa proporção reflete o comportamento operacional padrão do restaurante, de forma aproximada para fins de executarmos o desenvolvimento deste trabalho.

2. [Base de Dados de Avaliações \(Classificação de Satisfação\)](#): Esta base contém dados qualitativos coletados a partir da percepção da comunidade acadêmica sobre o cardápio diário, coletada informalmente por meio de grupos do *Whatsapp*, onde era enviada uma enquete para avaliação do prato do dia.

Data	Refeição	Campus	Cardápio	Nota 1	...	Nota 5	Avaliações	Média Bayesiana	Classe
------	----------	--------	----------	--------	-----	--------	------------	-----------------	--------

2024-04-03	Almoço	SM	FEIJOADA	1	...	38	56	4.55	Ótima
2024-04-01	Almoço	SM	CREME DE FRANGO	1	...	26	62	4.12	Boa
2024-04-04	Almoço	SM	BIFE SUÍNO ACEBOLADO	4	...	21	63	3.51	Aceitável
2024-04-02	Jantar	SM	LASANHA DE CARNE À BOLONHESA	28	...	12	56	2.80	Ruim
2024-08-17	Almoço	SM	ISCA DE FRANGO	0	...	0	0	Vazio	Sem Avaliação

- **Atributos e Tipos de Dados:** Os atributos de entrada mesclam dados categóricos textuais (Dia_Refeicao, Refeição, Campus, Cardápio) e atributos numéricos contínuos que representam a contagem de votos recebidos em cada nível de satisfação (Nota1 a Nota5 e o total de Avaliações).
- **Valores Ausentes:** Entradas referentes a dias sem funcionamento ou pratos sem nenhuma avaliação submetida geram contagens zeradas, que foram tratadas durante a fase de pré-processamento para evitar anomalias matemáticas.
- **Atributo Classe:** O atributo alvo é a CLASSE_AVALIACAO, segmentada em cinco categorias ordinais: *Péssima*, *Ruim*, *Aceitável*, *Boa* e *Ótima*, alinhando-se aos padrões de relatórios da Pró-reitoria de Assistência Estudantil (PROAE), que faz assim sua avaliação anual.
- **Distribuição e Balanceamento:** As avaliações de serviços de alimentação costumam apresentar forte viés positivo, o que naturalmente geraria um cenário de desbalanceamento extremo (poucas instâncias nas classes "Péssima" e "Ruim"). Para mitigar esse problema e auxiliar o aprendizado dos algoritmos, o atributo classe foi derivado a partir do ranqueamento relativo das notas no ecossistema do RU, forçando uma distribuição estatística mais equilibrada que garante suporte e representatividade para que o algoritmo aprenda os padrões de todas as cinco categorias.

Metodologia

Pré-processamento

Antes da aplicação dos algoritmos de classificação, foi realizado um processo detalhado de preparação das bases de dados, visando melhorar a qualidade das informações e garantir maior confiabilidade nos modelos gerados.

Inicialmente, foram identificados atributos redundantes e irrelevantes. Na base de avaliações das refeições, as notas individuais (Nota1 a Nota5) foram removidas, uma vez que a média bayesiana já representa de forma agregada e estatisticamente mais robusta a percepção dos usuários. O atributo Data também foi removido por não apresentar relação direta com a variável classe e possuir alta cardinalidade, o que poderia introduzir ruído no treinamento dos modelos.

Observou-se ainda a presença da classe “Sem avaliação” com apenas quatro instâncias. Por apresentar baixa representatividade e forte desbalanceamento em relação às demais classes, optou-se por remover essas instâncias utilizando o filtro RemoveWithValues do Weka, evitando viés nos algoritmos de classificação.

Em seguida, aplicou-se o filtro ReplaceMissingValues para tratar possíveis valores ausentes, substituindo-os automaticamente com base na distribuição dos dados. Essa etapa garantiu que todos os algoritmos pudessem ser executados corretamente, uma vez que muitos modelos de aprendizado supervisionado não lidam diretamente com valores faltantes.

Por fim, foi aplicado o filtro Normalize em todos os atributos numéricos. A normalização padroniza os valores em uma mesma escala, evitando que atributos com valores mais altos exerçam maior influência no treinamento, especialmente no algoritmo KNN, que é baseado em distância entre instâncias.

Essas etapas foram aplicadas de forma sequencial, garantindo consistência e qualidade na preparação dos dados.

Ferramenta utilizada

Para a execução dos experimentos foi utilizado o software **Weka (Waikato Environment for Knowledge Analysis)**, amplamente empregado em mineração de dados e aprendizado de máquina. O Weka permite a aplicação de algoritmos de classificação, pré-processamento de dados, avaliação de modelos e comparação de resultados de forma integrada.

Algoritmos

Foram selecionados três algoritmos clássicos de classificação supervisionada: J48, KNN (IBk) e Random Forest. A escolha desses algoritmos baseou-se em suas diferentes abordagens de aprendizado e ampla utilização em problemas de classificação.

O algoritmo J48, baseado em árvores de decisão, foi escolhido por sua capacidade de gerar modelos interpretáveis e lidar bem com atributos categóricos e numéricos. Árvores de decisão são amplamente utilizadas em problemas de classificação por permitirem fácil compreensão das regras geradas.

O algoritmo KNN (K-Nearest Neighbors) foi selecionado por ser um método baseado em instâncias e distância entre dados. Esse algoritmo permite avaliar a similaridade entre registros e verificar como a proximidade entre instâncias influencia na classificação. Sua inclusão possibilita comparar modelos baseados em estrutura de dados com modelos baseados em regras.

O Random Forest foi escolhido por ser um método ensemble baseado em múltiplas árvores de decisão. Esse algoritmo tende a apresentar maior robustez e capacidade de generalização, reduzindo problemas de overfitting e aumentando a precisão em diferentes cenários. Sua aplicação permite avaliar o ganho de desempenho obtido ao combinar múltiplos classificadores.

Além disso, o Random Forest foi o método que não vimos em sala. Ele é um conjunto de árvores de decisão em que cada árvore é treinada com um ruído aleatório específico, ou seja, cada árvore de decisão na floresta aleatória é treinada em um subconjunto diferente de exemplos. Random Forests são a forma mais conhecida de conjunto de árvores de decisão.

Estratégia de divisão de bases

Para a avaliação dos modelos foi utilizada a técnica de validação cruzada estratificada com 10 partições (10-fold cross-validation).

Nessa abordagem, o conjunto de dados é dividido em dez subconjuntos de tamanho aproximadamente igual. Em cada iteração, nove subconjuntos são utilizados para treinamento do modelo e o subconjunto restante é utilizado para teste. O processo é repetido dez vezes, garantindo que cada instância seja utilizada exatamente uma vez como dado de teste.

A validação cruzada estratificada preserva a proporção das classes em cada partição, mantendo a distribuição original dos dados durante o treinamento e teste. Essa característica é especialmente importante em problemas de classificação multiclasse, pois evita que alguma classe fique sub-representada em determinados subconjuntos.

A escolha do 10-fold cross-validation justifica-se por ser uma prática consolidada na literatura de aprendizado de máquina, oferecendo um bom equilíbrio entre custo computacional e confiabilidade estatística. Essa estratégia reduz a variabilidade da estimativa de desempenho e diminui o risco de superestimação dos resultados, quando comparada à simples divisão treino-teste.

Medidas de avaliação

As principais métricas utilizadas para avaliação dos modelos foram:

- **Acurácia (Accuracy):** porcentagem de instâncias corretamente classificadas.

$$Acurácia = \frac{Previsões\ Corretas}{Total\ de\ Previsões}$$

- **Kappa:** mede o grau de concordância entre as classificações previstas e reais, considerando o acaso.

$$K = \frac{P_o - P_e}{1 - P_e}$$

- **Precision, Recall e F-measure:** utilizadas para análise detalhada do desempenho por classe.

$$Precisão = \frac{Positivo\ Verdadeiro}{Positivo\ verdadeiro + Positivo\ Falso}$$

$$Recall = \frac{Positivo\ Verdadeiro}{Positivo\ Verdadeiro + Negativo\ falso}$$

$$F - Measure = \frac{2 * Precisão * Recall}{Precisão + recall}$$

A acurácia foi utilizada como principal métrica de comparação entre os modelos.

Parâmetros usados

Para cada algoritmo foram testadas diferentes configurações:

J48

- ConfidenceFactor = 0.25 (padrão)
- ConfidenceFactor = 0.15

KNN

- K = 3

- $K = 7$

Random Forest

- 100 árvores
- 50 árvores

Discussão de Resultados

Os resultados obtidos demonstraram diferenças relevantes de desempenho entre os algoritmos e entre as duas bases analisadas. Na base de avaliações das refeições, o algoritmo Random Forest apresentou o melhor desempenho geral, com acurácia próxima de 85%, seguido pelo J48 com desempenho semelhante. O algoritmo KNN apresentou desempenho significativamente inferior, com acurácia inferior a 60%.

Observou-se que a alteração do parâmetro *confidenceFactor* do J48 de 0,25 para 0,15 não produziu variação nos resultados obtidos. Esse comportamento indica que a árvore gerada na configuração padrão já apresentava bom nível de generalização e não possuía indícios relevantes de overfitting, não sendo necessária poda adicional. Dessa forma, a mudança do parâmetro não impactou a estrutura final do modelo nem sua capacidade de classificação.

Esse comportamento pode ser explicado pela natureza multiclasse da base de avaliações, que possui maior número de categorias e maior similaridade entre elas. Como o KNN baseia-se na distância entre instâncias, a sobreposição entre classes dificulta a separação adequada, reduzindo a precisão do modelo.

Na base de consumo de refeições, todos os algoritmos apresentaram desempenho superior, com acurácia acima de 87%. O Random Forest novamente apresentou o melhor resultado, alcançando aproximadamente 91,6% de acurácia. O KNN apresentou desempenho elevado nessa base, diferentemente da base anterior, indicando melhor separação entre as classes.

A melhoria no desempenho geral pode ser atribuída ao menor número de classes e à maior relação direta entre os atributos e a variável alvo. Isso facilita a identificação de padrões pelos algoritmos e reduz a ambiguidade na classificação.

Tabela 1 – Comparação de desempenho dos algoritmos nas bases analisadas

Observa-se que algoritmos baseados em árvores (J48 e Random Forest) apresentaram desempenho consistente em ambas as bases, demonstrando maior robustez frente a diferentes estruturas de dados. O Random Forest destacou-se como o modelo mais eficiente, apresentando maior capacidade de generalização.

Base	Algoritmo	Melhor Configuração	Accuracy
Avaliação das refeições	J48	ConfidenceFactor=0.25	84,54%
Avaliação das refeições	KNN	K = 3	57,48%
Avaliação das refeições	Random Forest	100 arvores	85,02%
Consumo de refeições	J48	ConfidenceFactor=0.25	90,24%
Consumo de refeições	KNN	K = 3	89,19%
Consumo de refeições	Random Forest	100 arvores	91,63%

Uso de LLMs

PROMPT:

Para execução de um trabalho preciso remodelar duas bases de dados. Para seguir as orientações do trabalho (pdf em anexo), preciso que essa data base tenha uma coluna com atributo classe.

Minha ideia era segmentar em 3 categorias de demanda: ALTA, NORMAL, BAIXA. para que os algoritmos pudessem ser usados nela como base de classificação (foco do principal do trabalho).

Mas, para além de segmentar nessas 3 categorias, gostaria que fosse considerado o dia da semana como base, por exemplo, a demanda normal de segunda é diferente da demanda de sexta, e quero levar isso em consideração. A princípio quero arrumar a base pelo google sheets, que possui uma função que converte date para dia da semana.

Também inclua que nessa fórmula seja considerado se é almoço ou jantar, assim, podemos unir a base de dados de almoço e jantar com uma coluna de flag dizendo qual a refeição.

Como eu poderia fazer essa segmentação para levar em consideração a demanda por dia da semana na equação?

RESPOSTA IA: Modelo: Gemini 3.1 PRO

Essa é uma jogada de mestre! Unir as duas bases de dados (almoço e jantar) vai aumentar a quantidade de linhas do seu conjunto de dados, o que ajuda muito os algoritmos de Machine Learning a aprenderem padrões de forma mais generalizada e robusta (evitando o *overfitting*). Como o comportamento do almoço é muito diferente do jantar, sua intuição está certíssima: precisamos calcular os percentis filtrando **ao mesmo tempo** pelo dia da semana e pelo tipo de refeição. Para fazer isso, basta adicionar uma segunda condição dentro da função **FILTER** no Google Sheets.

Passo a Passo da Nova Estrutura

Vamos assumir a seguinte organização de colunas na sua nova base unificada:

- **Coluna G:** **QNT_TOTAL** (A demanda que queremos classificar)
- **Coluna H:** **TIPO_REFEICAO** (Aqui você preenche com "Almoço" ou "Jantar")
- **Coluna I:** **DIA_SEMANA** (A fórmula **=DIA.DA.SEMANA(A2; 2)**)
- **Coluna J:** **CLASSE_DEMANDA** (A nossa nova fórmula)

Na célula **J2** (supondo que os dados começam na linha 2), insira a seguinte fórmula e arraste para baixo:

```
=IFS(  
  G2 >= PERCENTIL(FILTER(G:G; I:I=I2; H:H=H2); 0,75); "ALTA";  
  G2 <= PERCENTIL(FILTER(G:G; I:I=I2; H:H=H2); 0,25); "BAIXA";  
  VERDADEIRO; "NORMAL"  
)
```

Dessa forma, se a linha atual for uma "Terça-feira" no "Almoço", a fórmula vai calcular os 25% maiores e menores usando **apenas** o histórico de almoços das terças-feiras!

PROMPT:

Me ajude a organizar minhas anotações acima para melhorar a congruência e semântica entre os tópicos que anotamos sobre o pré-processamento e configurações do Weka?

""ReplaceMissingValues(weka)--> Para substituir os espaços vazios e permitir que os algoritmos funcionem.

Normalize(weka) --> Como usaremos o algoritmo KNN que é baseado em distância, a normalização foi aplicada para evitar que atributos com maior escala influenciassem mais no cálculo da distância.

RemoveWithValues(weka) --> Foi observado que a classe "Sem avaliação" possuía apenas 4 instâncias, caracterizando forte

desbalanceamento e baixa representatividade. Dessa forma, optou-se por remover essas instâncias para evitar viés nos algoritmos de classificação

Remove(weka) -->Foram removidos atributos individuais de notas (Nota1 a Nota5) e o atributo Data, pois a média bayesiana já representa de forma agregada as avaliações. A remoção desses atributos evitou redundância e reduziu o ruído nos dados, melhorando a qualidade do treinamento dos algoritmos.

Inicialmente foram removidos atributos considerados redundantes. Em seguida, aplicou-se o filtro ReplaceMissingValues. Por fim, os atributos numéricos foram normalizados para padronização da escala. “””

RESPOSTA IA: Modelo: Gemini 3.1 PRO

Aqui está uma proposta para organizar suas anotações em um texto coeso, fluido e com vocabulário acadêmico, ideal para compor a seção de **Pré-processamento** do seu relatório final.

Estruturei o texto seguindo a ordem cronológica que você mencionou no final das suas anotações:

Pré-processamento dos Dados (Ferramenta Weka)

A etapa de pré-processamento foi realizada na ferramenta Weka, seguindo uma esteira lógica de transformações com o objetivo de limpar, padronizar e adequar a base de dados para o treinamento dos algoritmos de classificação. O processo foi dividido nas seguintes etapas sequenciais:

1. Limpeza de Redundâncias e Tratamento de Desbalanceamento

Inicialmente, realizou-se a remoção de atributos e instâncias que poderiam introduzir ruídos ou enviesar os modelos:

- **Filtro Remove:** Foram excluídos os atributos individuais de contagem de notas (Nota1 a Nota5) e o atributo Data. Essa decisão baseou-se no fato de que o atributo *Média Bayesiana* já encapsula a representação agregada dessas avaliações. A remoção dessas colunas evitou a redundância de informações e o vazamento de dados, melhorando significativamente a qualidade do espaço de características para os algoritmos.
- **Filtro RemoveWithValues:** Observou-se que a classe “Sem avaliação” possuía apenas 4 instâncias, o que caracterizava uma baixíssima representatividade e um forte desbalanceamento. Para evitar que os classificadores criassem um viés na

tentativa de aprender um padrão com base em uma amostragem tão pequena, optou-se por aplicar este filtro para remover essas instâncias específicas.

2. Tratamento de Valores Ausentes

- **Filtro `ReplaceMissingValues`:** Após a limpeza inicial, aplicou-se este filtro para tratar eventuais espaços vazios (dados faltantes) residuais na base. A substituição desses valores por medidas estatísticas centrais é um requisito fundamental para viabilizar a execução matemática e o funcionamento correto da maioria dos algoritmos preditivos.

3. Padronização de Escala

- **Filtro `Normalize`:** Por fim, os atributos numéricos foram normalizados. Essa etapa foi indispensável devido à utilização de algoritmos baseados em cálculo de distância euclidiana, como o KNN (IBk). A normalização garante que todas as variáveis contínuas sejam redimensionadas para uma escala padronizada (geralmente entre 0 e 1), evitando que atributos com amplitudes originais maiores dominem ou influenciam desproporcionalmente o cálculo das distâncias durante o treinamento do modelo.

Dica para o relatório: Essa estrutura em tópicos deixa muito claro que você não apenas apertou os botões no Weka, mas que entendeu a justificativa matemática e estratégica por trás de cada filtro aplicado na ferramenta!