PROJETO APRENDIZADO DE MÁQUINA

1. **Nome : Natan Lima Viana**

2. Resultados Obtidos

2.1. Descrição dos Classificadores

Os classificadores usados foram :

- A priori , no qual a classificação a estimativa para a classficação dada pra o filme em questão é simplesmente a média truncada das demais avaliações para esse mesmo filme, ou seja, a média arredondada para o inteiro mais próximo.
- Árvore de decisão recursiva, na qual os nós não folha poderiam ser age , gender , occupation , movieID ou qualquer um dos genres possíveis para os filmes. Zip code não foi considerado pra evitar overffiting.

2.2. Dados e Resultados da comparação

Após rodar o programa (o que levou quase 1h no meu computador), o resultado obtido foi o seguinte :

```
O classificador a priori retornou os seguintes resultados
Matriz de confusão
|0||0||1||0||0|
|0||0||1||1||0|
|0||0||2||0||0|
|0||1||0||1||0|
|0||0||0||3||0|
Taxa de acerto : 0.3
Estatística Kappa : 0.125
Erro Quadrático Médio  : 1.6
-----------------------------------------------------
O classificador arvore de decisão retornou os seguintes resultados
Matriz de confusão
|0||0||0||0||1|
|0||0||0||0||2|
|0||0||0||0||2|
|1||0||0||0||1|
|0||0||0||0||3|
Taxa de acerto : 0.3
Estatística Kappa : 0.125
Erro Quadrático Médio  : 5.1
-----------------------------------------------------
```

2.3. Discussão e sugestão de melhorias para o classificador

O classificador da árvore de decisão poderia adotar critérios de poda pra evitar overffiting. Pode-se observar que qualquer que seja o critério adotado, a árvore de decisão obteve um resultado pior ou igual ao da classificação a priori.

Talvez naive bayes também não obtivesse um resultado tão bom , pois os gêneros dos filmes não são completamente independentes (um documentário musical, por exemplo, é difícil de acontecer).

O cálculo do erro quadrático médio levou em conta que os classificadores são não enviezados e que o peso do erro é a menor distância para a diagonal principal.

Mais do que 10 avaliações talvez refletisse um resultado mais fiel a realidade, além disso, poderia se pensar num classificador que usasse as classificações anteriores do próorio usuário para fazer a estimativa.

O código também demorou bastante pra rodar, daria pra fazer algumas otimizações, armazenando listas usadas frequentemente, por exemplo, além de armazenas os dados de uma forma mais conveniente (num banco de dados).

3. Conclusões:

Mesmo com um espaço amostral desse tamanho, é difícil prever de maneira precisa a classificação usando apenas árvore de decisão, na vida real provavelmente seriam necessárias técnicas mais avançadas (algoritmo adaptativo etc), que mesmo assim não iriam garantir um sucesso completo, já que não é possível modelar perfeitamente o gosto de alguém tomando como base apenas algumas informações básicas.

Apesar disso, a tarefa foi bastante interessante para se familiarizar com alguns conceitos importantes relativos ao aprendizado de máquina (matriz de confusão , estatística kappa , classificadores , árvores de decisão etc). O erro talvez tenha acontecido pelo fato de o usuário ter um gosto não muito comum.

4. Descrição da Implementação:

A minha implementação para a árvore de decisão foi levemente diferente da mostrada em aula, pois em vez de gerar todos os ramos da árvore recursivamente e 1 só vez, eu gerei só as subárvores relacionadas ao par filme-usuário que estava sendo classificado, economizando assim processamento.

A classe classificador é onde ficou a main, ele fez a leitura dos arquivos e os armazenou em listas com os tipos Movie , Rating e User, que são classes do pacote tabelas .

Na main é chamado a árvore de decisão, que escolhe como melhor atributo corrente aquele que tem maior ganho , que é calculado em função da entropia, é chamado também o Comparador, que é quem gera a matriz de confusão e, por fim, o Formatador, para mostrar o resultado na tela. Por fim, a classe atributos guarda apenas umas lista de Strings com os atributos que ainda não foram escolhidos na execução corrente da árvore de decisão.