universidade de aveiro

Trabalho Prático de MPEI

Triagem no Hospital Nacional

Autores:

Afonso Cardoso : 88964

Márcia Pires : 88747

Professor Regente:

Prof. Carlos Bastos

8 Dezembro de 2018

Conteúdo

1 Lista de Módulos

- Counting Bloom Filter (BloomFilter.java)
- Min Hashing (minHashing.java)
- Gerador de Pessoas (RandomGenerator.java)
- Contador Estocástico (StochasticCounter.java)
- Classe que gera Utentes (User.java)
- Testes dos Módulos (TestBloomFilter.java, TestMinHashing.java e StochasticCounterTest.java)
- Ficheiro a executar (Triage.java)

2 Como executar a Triagem

2.1 Triage

A classe Triage é a classe principal de todo o projeto visto que é aqui que se se concretiza a entrada dos pacientes no hospital. Estes pacientes, sejam eles introduzidos manualmente ou gerados aleatoriamente, são passados para o BloomFilter e ali

3 Módulos Auxiliares

3.1 BloomFilter

O ficheiro do Bloom Filter, no contexto apresentado, visa a inserção de pacientes, sejam eles introduzidos pelo utilizador, ou gerados aleatoriamente, sendo passado como argumento o seu nome.

Nesta função de inserção, insert(), o nome do paciente sofre um processo de codificação através do cálculo do seu código hash e é aí então introduzido no Bloom Filter. Após introduzidos os dados, é facilmente possível verificar se, dado um paciente, este já pertence ou não ao Bloom Filter, ou seja, se já deu entrada no hospital ou não, através do método isMember().

De modo a obter um valor adequado de funções de hash, no construtor desta classe são feitos os cálculos necessários à otimização deste valor.

3.2 minHashing

O processo de minHashing presente no nosso projeto, tem como objetivo analisar os sintomas de um dado paciente e realizar os sintomas similares entre os sintomas do utente e os sintomas de 6 doenças pré-definidas no sistema do hospital.

Assim, através de um processo de minHash comparamos dois Arrays de Strings, sendo um deles o dos sintomas do paciente e o outro Array de Strings dos sintomas da doença a analisar no momento.

Como no nosso tema os sintomas são importantes e necessários de analisar na integra, não recorremos à utilização do método *getShingles()*, no entanto este é criado e desenvolvido neste mesmo Módulo.

3.3 RandomGenerator

Neste módulo estão presentes Arrays de Strings para os Nomes próprios e sobrenomes, bom como locais, sintomas individuais e as doenças presentes e avaliadas no hospital compostas por 5 sintomas principais que nos ajudaram a realizar o diagnóstico.

Para além destes, existe também o construtor que gera um utente do hospital com os campos anteriormente mencionados, para assim obtermos utentes a inserir no Bloom Filter.

3.4 StochasticCounter

O contador estocástico permite fazer o rastreio de um dado utente no hospital aquando da entrada deste no mesmo, analisando assim número total de entradas.

Através deste método e de uma probabilidade do utente estar doente cada vez que dá entrada, é possível obtermos o número aproximado que este efetivamente estava doente e foi diagnosticado.

3.5 User

A classe User, assemelha-se à ficha hospital de um dado utente com todas as informações necessárias para a avaliação do individuo e para que lhe seja feito o diagnóstico.

; É possível criar a sua própria ficha hospital introduzindo os dados do terminal, ou pode ser criado um utente aleatório com os dados já preenchidos recorrendo à função *GeneratePerson()* presente no Módulo *RandomGenerator*.

4 Testes dos Módulos

Em conjunto com os módulos auxiliares, existem também os seus respetivos testes onde estes são testados na sua própria *main* com valores já definidos, ou em alternativa com valores e entradas introduzidos pelo utilizador.

5 Manual para o Utilizador

Contextualizando o utilizador, este deve ser capaz de perceber que ao utilizador a classe principal, Triage, está a dar entrada num hospital à espera que lhe seja feita uma triagem. A entrada do utilizador pode ser efetuada de duas maneiras distintas: manualmente, sendo todos os parâmetros introduzidos pelo mesmo e efetuados através da escolha 1) do menu apresentado; ou pode ser gerado aleatoriamente, através da classe RandomGenerator, escolhendo assim a opção 2) do menu que não só gera a entrada de um paciente, mas sim a de 20 diferentes, por forma a criar um tipo de base de dados para que seja possível obter mais meios de comparação para fins estatísticos.

Criados os pacientes, se este foi introduzido manualmente, tem a escolha de, face aos seus sintomas, ser-lhe efetuado um diagnóstico. Este diagnóstico é então realizado através da procura da similaridade de tais sintomas com os sintomas e respetivas doenças já presentes na base de dados do hospital, sendo estas as mais comuns. Realizadas as contas, é possível então deduzir a doença mais provável que tenha trazido o paciente ao hospital.

Idealmente, através deste dado e, comparativamente a todos os outros gerados aleatoriamente, seria possível concluir estatísticas acerca de quais doenças são mais prováveis de levar os pacientes ao hospital.

6 Considerações Finais

Após realizar o projeto, ficamos sem a dúvida de que todos estes Módulos lecionados aquando da disciplina de Métodos Probabilísticos para Engenharia Informática estão presentes nas plataformas mais conhecidas e utilizadas pela sociedade, sendo assim estes conceitos muito úteis.

A nível da concretização, os métodos estão bem desenvolvidos e individualmente estes funcionam, porém aquando da invocação destes na classe *Triage*, algum erro acontece no cálculo da similaridade, provocando assim no nosso contexto, um erro no diagnóstico, tornando-o inconclusivo.