The Turingator's Used Cars Rater Application

The domain: - The application which I have built is for determining the usability of used cars. Dealers which sell used cars can have customers that ask various questions about the cars. And dealers can lie claiming things about the cars that aren't true. So to be fair to the customers this application can be used by the customers after they have test-driven the car. My application asks for inputs like: -

Mileage: - average miles/gallon. Although I know this cannot be determined by just driving the car once. But this parameter is the foremost one looked by the customers. This variable is fuzzified in my domain and there are three likely values to it: - bad, average, and good.

Cost: - The cost of the car. Whether is it a good buy for the amount of dollars spent on them? This variable is fuzzified in my domain and there are three values for it: - costly, somewhat-costly (affordable), cheap.

Safety: - The safety of the cars is one of the very crucial parameters. As the number of accidents are increasing every year, it is always a good investment to buy somewhat-costly safe cars than cheap unsafe cars. This variable is also fuzzified in my domain and there are two values to it: - risky, and safe. It did not make sense to have a third variable like "completely-unsafe" cars, as most of the cars do have one or two safety features like anti-lock and airbags.

Reputation of the brand: - This is also one of the very important parameter as the reputation of the car's brand plays an important role in people's preconceptions and decisions. This variable too is fuzzified in my domain and there are two values to it: - insignificant, and reputed.
So there are a total of 3*3*2*2 possible values as such there are 36 rules. Each of these rules are the antecedents to a particular rule and the conclusion is whether the car is a good buy or not.

The **conclusion** is also fuzzified and there are three likely values: - **bad car, somewhat-good car, and good car**. There are for obvious reasons many more parameters which are used by the customers to determine the worthiness of the used-cars namely, number of accessories, tire conditions, fuel-type, how old is the car, etc. But as the number of fuzzy variables increase, so do the rules. To keep the rules to limited number I have used 4 fuzzy-variables and 36 rules.

I have used completely Java code to implement my application. The grader needs to read through the README section of this document to see how to attach the required jars to the build path and how to run the application in the IDE or just as a jar from command-line. The grader can contact me any time if there is any problem while running the code or the jar.

I have used **PIFuzzySet** to model each of my fuzzy variables. The two attributes in the PIFuzzySet(2.0, 2.0) are: first is the mid-point of the curve, second term is the curve width. So 2.0 is the centre and spans 2.0 on left and 2.0 on right.

The **main()** function in the code is the entry point into the application. The user is then prompted to input values for the variables: - mileage, cost, safety, and brand reputation. Based on these values the crisp value is created and then rules are triggered. For each of the matching rule I am taking a union of the fuzzy sets. Later when all the matched rules are triggered I defuzzify the value using momentDefuzzify API.

This API is more useful than the maximumDefuzzify API because it says in the documentation that *"maximumDefuzzify finds the mean of the maximum values of the FuzzySet of the FuzzyValue as the defuzzification value. NOTE: This doesn't always work well because there can be x ranges where the y value is constant at the max value and other places where the max is only reached for a single x value. When this happens the single value gets too much of a say in the defuzzified value."*

Later based on the defuzzified value I determine whether the car is worth buying or not.

```
if(carRating >= 0.0 && carRating <= 4.0) {
      System.out.println("The car is in bad condition, do not waste hard
earned money!");
} else  if(carRating > 4.0 && carRating < 8.0) {
      System.out.println("The car is in somewhat-good condition, can buy!");
} else  if(carRating >= 8.0 && carRating <= 10.0) {
      System.out.println("The car is in very good condition, throw the money
and run!");
}
```

Please have a look at the syntax of my rules given below: -

```
//I have 36 rules in my system. Each rule takes in 4 antecedents and one
conclusion. The antecedents correspond to the fuzzy variables: - mileage,
cost, safety, reputation. And conclusion corresponds to the fuzzy variable
rating. There are a total of 3*3*2*2 values/terms in each of my antecedent
fuzzy variables as such there are 36 rules.
//Rule 1
carRatingAttribute[count] = new FuzzyRule();

carRatingAttribute[count].addAntecedent(new FuzzyValue(mileage,"low"));
carRatingAttribute[count].addAntecedent(new FuzzyValue(cost,"costly"));
carRatingAttribute[count].addAntecedent(new FuzzyValue(safety,"risky"));
carRatingAttribute[count].addAntecedent(new
FuzzyValue(reputation,"insignificant"));

carRatingAttribute[count++].addConclusion(new FuzzyValue(rating,"bad"));
```

Do let me know if you do not understand anything in my code or logic. I have commented my code well too.

README Section: -

The current project folder has 1 folder and 1 report pdf. The folder contains a "src" (./UsedCarRater/src/) folder which has the code "UsedCarRaterApplication.java". The grader can use any of the two mentioned procedures to run the application.

Procedure 1: -

- Please download the required jar from this link: -
  - https://github.com/rorchard/FuzzyJ/blob/master/fuzzyJ-2.0.jar
  - Create a new Eclipse IDE Project and create a new class and copy my code given in the "src". Please add the downloaded jar to the build path of the project in the IDE.
  - Right click onto the project name and go to **Properties** and go to **Java Build Path** then go to **Libraries** and then click on the **Add External JARs…** and give the location where you downloaded the above jar. The project will be free of errors now and can be run now.

Procedure2: -

- The grader can just go to the ./UsedCarRater/src/UsedCarRaterApplication.java source code and see the code and can then just run the attached jar from this Dropbox location, if the grader doesn't want to do the above procedure.
  - https://www.dropbox.com/s/i9h3g4lwafy18fv/UsedCarRaterApplication.jar?dl=0
  - Go to the command prompt or terminal if using a linux, go to the folder where the above jar was downloaded and run the command
  - **Java -jar UsedCarRaterApplication.jar**
  - The jar will run the application and prompt for the values.

Please have a look into the test cases attached here for the possible values for each of the variables.

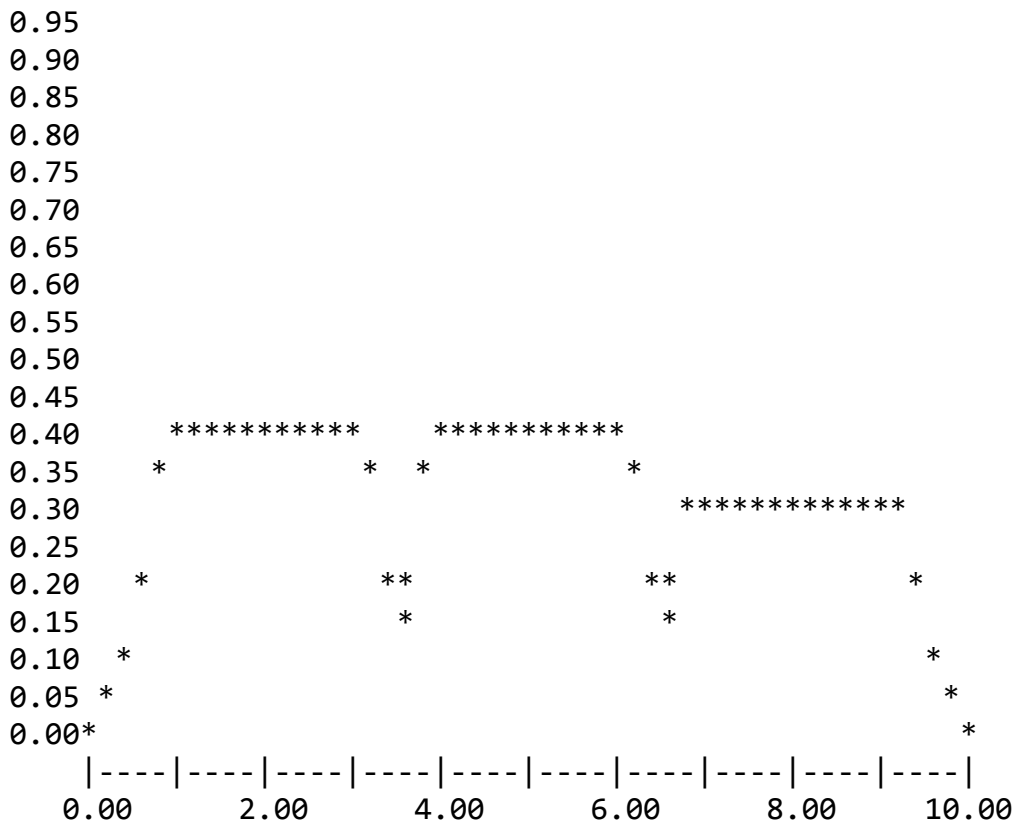
Test Case 1: -

```
Please enter value for car mileage:
low = 0.0 - 4.0, medium = 4.0 - 6.0, high = 6.0 - 10.0
6.57
Please enter value for car cost
costly = 0.0 - 4.0, affordable = 4.0 - 6.0, cheap = 6.0 - 10.0
3.45
Please enter value for car safety
risky = 0.0 - 5.0, safe = 5.0 - 10.0
6.28
Please enter value for brand reputation
insignificant = 0.0 - 5.0, reputed = 5.0 - 10.0
3.28
The car rating on a scale of 10 is : 5.0


MESSAGE: - The car is in somewhat-good condition, can buy!


Overall Car Assessment: -
FuzzyVariable         -> car rating [ 0.0, 10.0 ] On a Scale of 1-10 car
ratings of bad, somewhat-good, good
Linguistic Expression -> good
FuzzySet              -> { 0/6 0.12/6.5 0.5/7 0.88/7.5 1/8 0.88/8.5 0.5/9
0.12/9.5 0/10 }
Fuzzy Value: car rating
Linguistic Value: ??? (*)

 1.00
```

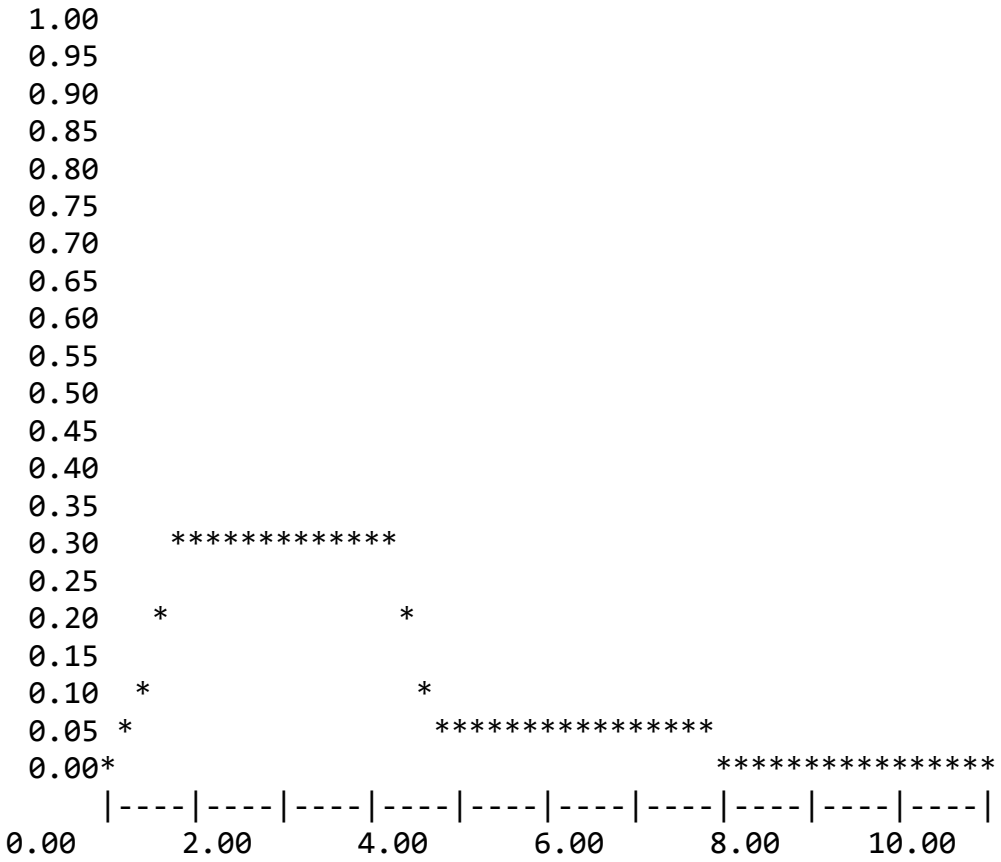


Test Case 2: -

```
Please enter value for car mileage:
low = 0.0 - 4.0, medium = 4.0 - 6.0, high = 6.0 - 10.0
3.4
Please enter value for car cost
costly = 0.0 - 4.0, affordable = 4.0 - 6.0, cheap = 6.0 - 10.0
2.7
Please enter value for car safety
risky = 0.0 - 5.0, safe = 5.0 - 10.0
1.2
Please enter value for brand reputation
insignificant = 0.0 - 5.0, reputed = 5.0 - 10.0
1.1
The car rating on a scale of 10 is : 2.0


MESSAGE: - The car is in bad condition, do not waste hard earned money!


Overall Car Assessment: -
FuzzyVariable         -> car rating [ 0.0, 10.0 ] On a Scale of 1-10 car
ratings of bad, somewhat-good, good
Linguistic Expression -> good
FuzzySet              -> { 0/6 0.12/6.5 0.5/7 0.88/7.5 1/8 0.88/8.5 0.5/9
0.12/9.5 0/10 }
Fuzzy Value: car rating
Linguistic Value: ??? (*)
```



Test Case 3: -

```
Please enter value for car mileage:
low = 0.0 - 4.0, medium = 4.0 - 6.0, high = 6.0 - 10.0
9.7
Please enter value for car cost
costly = 0.0 - 4.0, affordable = 4.0 - 6.0, cheap = 6.0 - 10.0
7.8
Please enter value for car safety
risky = 0.0 - 5.0, safe = 5.0 - 10.0
8
Please enter value for brand reputation
insignificant = 0.0 - 5.0, reputed = 5.0 - 10.0
9
The car rating on a scale of 10 is : 7.0


MESSAGE: - The car is in somewhat-good condition, can buy!


Overall Car Assessment: -
FuzzyVariable         -> car rating [ 0.0, 10.0 ] On a Scale of 1-10 car
ratings of bad, somewhat-good, good
Linguistic Expression -> good
FuzzySet              -> { 0/6 0.12/6.5 0.5/7 0.88/7.5 1/8 0.88/8.5 0.5/9
0.12/9.5 0/10 }
Fuzzy Value: car rating
```

```
Linguistic Value: ??? (*)

 1.00
 0.95
 0.90
 0.85
 0.80
 0.75
 0.70
 0.65
 0.60
 0.55
 0.50
 0.45
 0.40
 0.35
 0.30
 0.25                                  *************
 0.20                                 *             *
 0.15
 0.10                                *               *
 0.05                  ****************                 *
 0.00****************                                    *
     |----|----|----|----|----|----|----|----|----|----|
    0.00      2.00      4.00      6.00      8.00     10.00
```

Test Case 4: -

```
Please enter value for car mileage:
low = 0.0 - 4.0, medium = 4.0 - 6.0, high = 6.0 - 10.0
10
Please enter value for car cost
costly = 0.0 - 4.0, affordable = 4.0 - 6.0, cheap = 6.0 - 10.0
10
Please enter value for car safety
risky = 0.0 - 5.0, safe = 5.0 - 10.0
10
Please enter value for brand reputation
insignificant = 0.0 - 5.0, reputed = 5.0 - 10.0
10
The car rating on a scale of 10 is : 8.0


MESSAGE: - The car is in very good condition, throw the money and run!


Overall Car Assessment: -
FuzzyVariable         -> car rating [ 0.0, 10.0 ] On a Scale of 1-10 car
ratings of bad, somewhat-good, good
Linguistic Expression -> good
```

```
FuzzySet              -> { 0/6 0.12/6.5 0.5/7 0.88/7.5 1/8 0.88/8.5 0.5/9
0.12/9.5 0/10 }
Fuzzy Value: car rating
Linguistic Value: ??? (*)

 1.00
 0.95
 0.90
 0.85
 0.80
 0.75
 0.70
 0.65
 0.60
 0.55
 0.50
 0.45
 0.40
 0.35
 0.30
 0.25
 0.20
 0.15
 0.10                                  *****************
 0.05                                 *                 *
 0.00*******************************                   *
     |----|----|----|----|----|----|----|----|----|----|
    0.00      2.00      4.00      6.00      8.00     10.00
```

Test Case 5: -

```
Please enter value for car mileage:
low = 0.0 - 4.0, medium = 4.0 - 6.0, high = 6.0 - 10.0
8
Please enter value for car cost
costly = 0.0 - 4.0, affordable = 4.0 - 6.0, cheap = 6.0 - 10.0
9
Please enter value for car safety
risky = 0.0 - 5.0, safe = 5.0 - 10.0
8
Please enter value for brand reputation
insignificant = 0.0 - 5.0, reputed = 5.0 - 10.0
9
The car rating on a scale of 10 is : 7.0


MESSAGE: - The car is in somewhat-good condition, can buy!


Overall Car Assessment: -
FuzzyVariable         -> car rating [ 0.0, 10.0 ] On a Scale of 1-10 car
ratings of bad, somewhat-good, good
```

```
Linguistic Expression -> good
FuzzySet              -> { 0/6 0.12/6.5 0.5/7 0.88/7.5 1/8 0.88/8.5 0.5/9
0.12/9.5 0/10 }
Fuzzy Value: car rating
Linguistic Value: ??? (*)

 1.00
 0.95
 0.90
 0.85
 0.80
 0.75
 0.70
 0.65
 0.60
 0.55
 0.50
 0.45
 0.40
 0.35
 0.30
 0.25                                  *************
 0.20                                 *             *
 0.15
 0.10                                *               *
 0.05                  ****************                 *
 0.00*****************                                   *
     |----|----|----|----|----|----|----|----|----|----|
    0.00      2.00      4.00      6.00      8.00     10.00
```

As given before there are 3 terms for mileage, 3 for cost, 2 for safety, and 2 for brand reputation. Please give the values between 0 – 10.

Do let me know if you have any difficulties while running the application or any kind of problems. Thank you for your time.
