# Challenge April-2021
## Rate Loan Risk Category

### A solution with DT5GL by Jack Jansonius – 12 April 2021

An interesting decision model because I cannot simply adopt it into my own tool.
The reason is that the proposed model is based on data-driven, forward reasoning, while my own model is based on goal-driven, backward reasoning. A goal-driven approach requires additional rules, and that is what I want to show with this solution. In addition, I want to indicate the benefits of a goal-driven approach over a data-driven approach.

First, the proposed decision model of the challenge:

LTV Risk
*tRiskCategory*
*"Low", "Medium", "High"*

| | inputs | outputs | annotations |
|---|---|---|---|
| U | Loan Summary.LTVPct | LTV Risk | Description |
| | *Number* | *tRiskCategory* *"Low", "Medium", "High"* | |
| 1 | <60 | "Low" | |
| 2 | [60..80] | "Medium" | |
| 3 | >80, null | "High" | |

Loan Risk
*tRiskCategory*
*"Low", "Medium", "High"*

| | inputs | | | outputs |
|---|---|---|---|---|
| P | Worst DTI | LTV Risk | Worst Credit Score | Loan Risk |
| | *tPercent* | *tRiskCategory* *"Low", "Medium", "High"* | *Number* | *tRiskCategory* *"Low", "Medium", "High"* |
| 1 | <36 | "Low" | >=680 | "Low" |
| 2 | <36 | "Medium" | >700 | "Low" |
| 3 | <36 | "Low" | [640..680) | "Medium" |
| 4 | <36 | "Medium" | [660..700] | "Medium" |
| 5 | [36..45] | "Low" | >=680 | "Medium" |
| 6 | [36..45] | "Medium" | >700 | "Medium" |
| 7 | - | - | - | "High" |

Data-driven, the first decision table for determining the variable LTV_Risk will be run anyway, for 2 reasons. First, because this table appears first in the model, and second, because the variable LTV_Risk determined here is an input variable for the next decision table.

Nevertheless, in a number of cases it is not necessary to execute this first table first; from the second table it is easy to deduce that if Worst_DTI > 45 the conclusion can be drawn directly that the Loan_Risk = "High"; for this the values of the other input variables are not relevant.

Thus, if Worst_DTI > 45, the first table will be executed for no purpose.
This is problematic for several reasons.

First of all, executing the first table will retrieve (or require) information that is not relevant to the problem solving (which in fact implies non-intelligent behavior!). If the model is applied in an interactive environment, e.g. a web form, the user will be asked for the LTV first, while the question for the DTI is much more relevant. If the user answers the latter question truthfully 46, the question about the LTV would not have been asked at all.

This gathering of unnecessary information is made even worse if this first table has input variables that in turn must be (or have been) determined in other decision tables.

The way in which it is inferred in the second table that “Loan_Risk is High" makes this relevance problem even worse at the time when this fact appears again as a condition in other decision tables (as the condition "LTV_Risk = Medium" in the second table is determined in the first table).

Here we run into the ambiguous meaning of the horizontal dashes in the decision rule by which Loan_Risk = "High" is determined in the second table.

The only meaning that the trailing dashes in decision tables should have is "Not relevant" (or "Not possible", which is a special case of "Not relevant").

And then it turns out that 7 explicit rules are required for the conclusion “LoanRisk is High”, rather than one implicit rule in a data-driven approach:

| If: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Worst_DTI < 36 | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | N | N |
| Worst_DTI <= 45 | - | - | - | - | - | - | - | Y | Y | Y | Y | Y | N |
| LTV_Risk is Low | Y | Y | Y | N | N | N | N | Y | Y | N | N | N | - |
| LTV_Risk is Medium | - | - | - | Y | Y | Y | N | - | - | Y | Y | N | - |
| Worst_Credit_Score >= 680 | Y | N | N | - | - | - | - | Y | N | - | - | - | - |
| Worst_Credit_Score >= 640 | - | Y | N | - | - | - | - | - | - | - | - | - | - |
| Worst_Credit_Score > 700 | - | - | - | Y | N | N | - | - | - | Y | N | - | - |
| Worst_Credit_Score >= 660 | - | - | - | - | Y | N | - | - | - | - | - | - | - |
| Then: | | | | | | | | | | | | | |
| LoanRisk is Low | X | | | X | | | | | | | | | |
| LoanRisk is Medium | | X | | | X | | | X | | X | | | |
| LoanRisk is High | | | X | | | X | X | | X | | X | X | X |

Thus, querying conditions and executing required decision tables in order of relevance requires goal-driven processing of the tables that are provided with explicit rules.

Based on the full table, the explicit rules for deriving “LoanRisk is High” can also be displayed horizontally:

| Worst DTI | LTV Risk | Worst Credit Score | Loan Risk |
|---|---|---|---|
| <36 | Low | <640 | High |
| <36 | Medium | <660 | High |
| <36 | High | - | High |
| [36..45] | Low | <680 | High |
| [36..45] | Medium | <=700 | High |
| [36..45] | High | - | High |
| >45 | - | - | High |

Now the dashes in the table do have the meaning "Not Relevant" and therefore this table can also be processed in a goal driven way!

And that means that the first table for determining LTV Risk is not executed if Worst DTI is greater than 45.

See further my solution to the June 2017 Loan Origination Challenge:
https://dmcommunity.org/challenge/challenge-june-2017/

On the following pages, I first show an interactive version of the solution, where the desired input variables are requested from the user.

With a slight modification of this interactive version, the desired input variables are then read from a SQLite database file and the results are then stored in that same file.

## Implementation of the decision tables in DT5GL; interactive version:

```
Table 0:
If:                                          | 0| 1|
'Continue'                                   | Y| N|
Then:
LoanRisk is Exit                             |  | X|
# .......

Proposition: 'Continue'
Askable_using: "Compute loan risk for next borrower?"

Table 1:
If:                           | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|
Worst_DTI < 36                | Y| Y| Y| Y| Y| Y| Y| N| N| N| N| N| N|
Worst_DTI <= 45               | -| -| -| -| -| -| -| Y| Y| Y| Y| Y| N|
LTV_Risk is Low               | Y| Y| Y| N| N| N| N| Y| Y| N| N| N| -|
LTV_Risk is Medium            | -| -| -| Y| Y| Y| N| -| -| Y| Y| N| -|
Worst_Credit_Score >= 680     | Y| N| N| -| -| -| -| Y| N| -| -| -| -|
Worst_Credit_Score >= 640     | -| Y| N| -| -| -| -| -| -| -| -| -| -|
Worst_Credit_Score >  700     | -| -| -| Y| N| N| -| -| -| Y| N| -| -|
Worst_Credit_Score >= 660     | -| -| -| -| Y| N| -| -| -| -| -| -| -|
Then:
LoanRisk is Low               | X|  |  | X|  |  |  |  |  |  |  |  |  |
LoanRisk is Medium            |  | X|  |  | X|  |  | X|  | X|  |  |  |
LoanRisk is High              |  |  | X|  |  | X| X|  | X|  | X| X| X|
# .......


Table 2:
If:                           | 0| 1| 2| 3|
LTV_Pct_provided is True      | Y| Y| Y| N|
LTV_Pct <  60                 | Y| N| N| -|
LTV_Pct <= 80                 | -| Y| N| -|
Then:
LTV_Risk is Low               | X|  |  |  |
LTV_Risk is Medium            |  | X|  |  |
# .......


Attribute: Worst_DTI
Askable_using: "What is the maximum provided Debt-To-Income ratio (DTI) of the
borrower [36-45]?"

Attribute: Worst_Credit_Score
Askable_using: "What is the minimum provided Credit Score of the borrower (600-
800)?"

Attribute: LTV_Pct_provided
Askable_using: "Loan-to-Value ratio (LTV) related to the desired loan amount
provided?"

Attribute: LTV_Pct
Askable_using: "What is the Loan-to-Value ratio (LTV) [60-80]?"
```

```
GoalAttribute: LoanRisk
Repeat_until: Exit

Case: Exit
Print: "Finished"

Case: Low
Print: "--------------------"
Print: "Loan Risk is: LOW"
Print: "--------------------"

Case: Medium
Print: "--------------------"
Print: "Loan Risk is: MEDIUM"
Print: "--------------------"

Case: High
Print: "--------------------"
Print: "Loan Risk is: HIGH"
Print: "--------------------"
```

## Some test cases:

```
"Compute loan risk for next borrower? (y/n)? > y
"What is the maximum provided Debt-To-Income ratio (DTI) of the borrower [36-45]?"
(int)> 35


"Loan-to-Value ratio (LTV) related to the desired loan amount provided?"
1. True
2. False
> 1


"What is the Loan-to-Value ratio (LTV) [60-80]?"
(int)> 59


"What is the minimum provided Credit Score of the borrower (600-800)?"
(int)> 680

--------------------
Loan Risk is: LOW
--------------------

"Compute loan risk for next borrower? (y/n)? > y
"What is the maximum provided Debt-To-Income ratio (DTI) of the borrower [36-45]?"
(int)> 46

--------------------
Loan Risk is: HIGH
--------------------

"Compute loan risk for next borrower? (y/n)? > n
Finished
```

## Implementation of the decision tables in DT5GL; database version:

```
Table 0:
If:                                          | 0| 1|
'Next borrower'                              | Y| N|
Then:
LoanRisk is Exit                             |  | X|
# .......

SQLite_database: "Database/Risk.db"

Proposition: 'Next borrower'
Obtain_instance_from_database_view: Borrower

Table 1:
If:                           | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|
Worst_DTI < 36                | Y| Y| Y| Y| Y| Y| Y| N| N| N| N| N| N|
Worst_DTI <= 45               | -| -| -| -| -| -| -| Y| Y| Y| Y| Y| N|
LTV_Risk is Low               | Y| Y| Y| N| N| N| N| Y| Y| N| N| N| -|
LTV_Risk is Medium            | -| -| -| Y| Y| Y| N| -| -| Y| Y| N| -|
Worst_Credit_Score >= 680     | Y| N| N| -| -| -| -| Y| N| -| -| -| -|
Worst_Credit_Score >= 640     | -| Y| N| -| -| -| -| -| -| -| -| -| -|
Worst_Credit_Score >  700     | -| -| -| Y| N| N| -| -| -| Y| N| -| -|
Worst_Credit_Score >= 660     | -| -| -| -| Y| N| -| -| -| -| -| -| -|
Then:
LoanRisk is Low               | X|  |  | X|  |  |  |  |  |  |  |  |  |
LoanRisk is Medium            |  | X|  |  | X|  |  | X|  | X|  |  |  |
LoanRisk is High              |  |  | X|  |  | X| X|  | X|  | X| X| X|
# .......


Table 2:
If:                           | 0| 1| 2| 3|
LTV_Pct_provided = Yes        | Y| Y| Y| N|
LTV_Pct <  60                 | Y| N| N| -|
LTV_Pct <= 80                 | -| Y| N| -|
Then:
LTV_Risk is Low               | X|  |  |  |
LTV_Risk is Medium            |  | X|  |  |
# .......


Attribute: Worst_DTI
Obtain_value_from_database_view: Borrower.DTIPct

Attribute: Worst_Credit_Score
Obtain_value_from_database_view: Borrower.CreditScore

Attribute: LTV_Pct_provided Type: Integer
Equals: int(LTV_Pct != None)
# 1 means: provided/not Null; 0 means: not provided/Null

Attribute: Yes Type: Integer
Equals: 1

Attribute: LTV_Pct
Obtain_value_from_database_view: Borrower.LTVPct
```

```
Database_view: Borrower
With_attributes:
Name, LoanAmount, LTVPct, CreditScore, DTIPct, LoanRisk
Query:
SELECT *
  FROM Borrower
 LIMIT 1 OFFSET %s
With_arguments: Borrower.auto_index


GoalAttribute: LoanRisk
Repeat_until: Exit

Case: Exit
Print: "Finished"

Case: Low
Print: "Loan Risk for %s: LOW"     Borrower.Name
>SQL:  "UPDATE Borrower "
-SQL:  "   SET LoanRisk = 'Low'    "
<SQL:  " WHERE Name = '%s' "       Borrower.Name

Case: Medium
Print: "Loan Risk for %s: MEDIUM"  Borrower.Name
>SQL:  "UPDATE Borrower "
-SQL:  "   SET LoanRisk = 'Medium' "
<SQL:  " WHERE Name = '%s' "       Borrower.Name

Case: High
Print: "Loan Risk for %s: HIGH"    Borrower.Name
>SQL:  "UPDATE Borrower "
-SQL:  "   SET LoanRisk = 'High'   "
<SQL:  " WHERE Name = '%s' "       Borrower.Name


Initial_database_setup: delete_borrowers
Query:
    DELETE FROM Borrower
End_Query

Initial_database_setup: insert_new_borrowers
Query:
    INSERT INTO Borrower
     (Name, LoanAmount, LTVPct, CreditScore, DTIPct)
    VALUES
    ('TC0',      20000,     59,         680,     35),
    ('TC1',      20000,     59,         679,     34),
    ('TC2',      20000,     59,         639,     35),
    ('TC3',      20000,     60,         701,     35),
    ('TC4',      20000,     80,         700,     35),
    ('TC5',      20000,     60,         659,     35),
    ('TC6a',     20000,     81,        Null,     35),
    ('TC6b',     20000,   Null,        Null,     35),
    ('TC7',      20000,     59,         680,     45),
    ('TC8',      20000,     59,         679,     36),
    ('TC9',      20000,     60,         701,     45),
    ('TC10',     20000,     60,         700,     44),
    ('TC11a',    20000,     81,        Null,     45),
    ('TC11b',    20000,   Null,        Null,     45),
    ('TC12',     20000,   Null,        Null,     46)
End_Query
```

## Testrun for the database-version:

```
Loan Risk for TC0: LOW
Loan Risk for TC1: MEDIUM
Loan Risk for TC2: HIGH
Loan Risk for TC3: LOW
Loan Risk for TC4: MEDIUM
Loan Risk for TC5: HIGH
Loan Risk for TC6a: HIGH
Loan Risk for TC6b: HIGH
Loan Risk for TC7: MEDIUM
Loan Risk for TC8: HIGH
Loan Risk for TC9: MEDIUM
Loan Risk for TC10: HIGH
Loan Risk for TC11a: HIGH
Loan Risk for TC11b: HIGH
Loan Risk for TC12: HIGH
Finished
```

| | Name | LoanAmount | LTVPct | CreditScore | DTIPct | LoanRisk |
|---|---|---|---|---|---|---|
| 1 | TC0 | 20000 | 59 | 680 | 35 | Low |
| 2 | TC1 | 20000 | 59 | 679 | 34 | Medium |
| 3 | TC2 | 20000 | 59 | 639 | 35 | High |
| 4 | TC3 | 20000 | 60 | 701 | 35 | Low |
| 5 | TC4 | 20000 | 80 | 700 | 35 | Medium |
| 6 | TC5 | 20000 | 60 | 659 | 35 | High |
| 7 | TC6a | 20000 | 81 | NULL | 35 | High |
| 8 | TC6b | 20000 | NULL | NULL | 35 | High |
| 9 | TC7 | 20000 | 59 | 680 | 45 | Medium |
| 10 | TC8 | 20000 | 59 | 679 | 36 | High |
| 11 | TC9 | 20000 | 60 | 701 | 45 | Medium |
| 12 | TC10 | 20000 | 60 | 700 | 44 | High |
| 13 | TC11a | 20000 | 81 | NULL | 45 | High |
| 14 | TC11b | 20000 | NULL | NULL | 45 | High |
| 15 | TC12 | 20000 | NULL | NULL | 46 | High |