**Credit Risk Analytics Challenge**

In this challenge, you’ll act as a **credit risk analyst** at **Nova Bank**, a financial institution that provides personal, medical, education, and business loans across the USA, UK, and Canada. Nova Bank wants to make lending fair and accessible while also protecting itself from unnecessary risk.

The main challenge is finding the right balance. If Nova Bank approves too many high-risk loans, it loses money. If it becomes too strict, it misses out on potential customers. By looking at the data, your job is to help the bank understand **who tends to default and why**, and how lending decisions can be made more reliable.

**What You’ll Do**

Use the dataset to build a short analysis or dashboard that helps Nova Bank:

* See which groups of borrowers are more or less likely to default
* Identify the factors that matter most when predicting loan outcomes
* Explore how loan size, income, interest rates, and repayment terms affect risk
* Spot early signs of financial trouble so action can be taken sooner
* Suggest ways the bank can adjust lending policies to be both safer and fairer

**Questions to Explore**

* Which types of borrowers are more likely to default?
* Do certain loan purposes (education, medical, personal, debt consolidation) carry more risk?
* How do **loan-to-income** and **debt-to-income ratios** relate to repayment?
* Does employment type or home ownership make a difference?
* How do past defaults or longer credit histories affect loan outcomes?
* Are there clear differences between borrowers in the USA, UK, and Canada?
* Which loan grades or terms seem safer, and which are riskier?
* Can groups of borrowers be identified that look “safe” versus “risky”?

✨ **Note:** These questions are just starting points. You’re encouraged to explore the data in your own way and share any unexpected findings that could help Nova Bank lend more responsibly.

### SWITCH Syntax in DAX

The **SWITCH** function has two primary forms: one for evaluating an expression against specific values and another for evaluating multiple true/false conditions (like **IF**).

#### 1. Basic SWITCH Syntax (Value-Based)

text

SWITCH(<expression>, <value1>, <result1>, <value2>, <result2>, ..., [<else>])

* **<expression>**: The value or column to evaluate (e.g., cb\_person\_cred\_hist\_length).
* **<value1>, <result1>**: If <expression> equals <value1>, return <result1>.
* **<else>**: Optional default result if no values match.
* **Returns**: The result corresponding to the first matching value, or the <else> result if no match.

**Example** (from your dataset): To categorize cb\_person\_cred\_hist\_length into buckets (as used in the prior analysis):

text

Credit History Bucket =

SWITCH(

'Credit Risk Data'[cb\_person\_cred\_hist\_length],

2, "2 Years",

3, "3 Years",

4, "4 Years",

5, "5 Years",

">5 Years"

)

* If cb\_person\_cred\_hist\_length = 2, returns "2 Years"; if 3, returns "3 Years"; else, returns ">5 Years".
* **Limitation**: This works for exact matches, so it’s less flexible for ranges (e.g., 2–3 years).

#### 2. SWITCH with TRUE() Syntax (Condition-Based)

For ranges or complex conditions (more relevant to your dataset), use **SWITCH** with **TRUE()**:

text

SWITCH(

TRUE(),

<condition1>, <result1>,

<condition2>, <result2>,

...,

[<else>]

)

* **TRUE()**: Evaluates conditions in order until one is true, returning the corresponding result.
* **<condition1>, <result1>**: If <condition1> is true, return <result1>.
* **<else>**: Optional default result if no conditions are true.
* **Returns**: The result of the first true condition, or <else> if none are true.

**Example** (from your dataset for loan\_to\_income\_ratio):

text

Loan-to-Income Bucket =

SWITCH(

TRUE(),

'Credit Risk Data'[loan\_to\_income\_ratio] <= 0.2, "<= 0.2",

'Credit Risk Data'[loan\_to\_income\_ratio] <= 0.4, "0.2 - 0.4",

'Credit Risk Data'[loan\_to\_income\_ratio] <= 0.6, "0.4 - 0.6",

"> 0.6"

)

* Evaluates conditions sequentially:
  + If loan\_to\_income\_ratio ≤ 0.2, returns "<= 0.2".
  + Else, if ≤ 0.4, returns "0.2 - 0.4".
  + Else, if ≤ 0.6, returns "0.4 - 0.6".
  + Else, returns "> 0.6".
* **Use Case**: Ideal for bucketing continuous values like ratios or cb\_person\_cred\_hist\_length, as seen in your prior queries.