

sas **innovate**
2025

Getting Started with SAS Intelligent Decisioning

Hands-on Workshop

Grace Barnhill, Technical Training Consultant
SAS EDU Data Management Content Development

Copyright © SAS Institute Inc. All rights reserved.

SAS® Intelligent Decisioning

Decisions are a combination of rule sets, analytical models, conditional logic, custom code, and other objects which are used to evaluate data and make real-world business decisions based on SAS decision output.



Should a mortgage loan be approved?



Is a transaction fraudulent?



Is a machine about to fail?



Is a medical claim valid?



Which offer should be sent to a customer?

Copyright © SAS Institute Inc. All rights reserved.

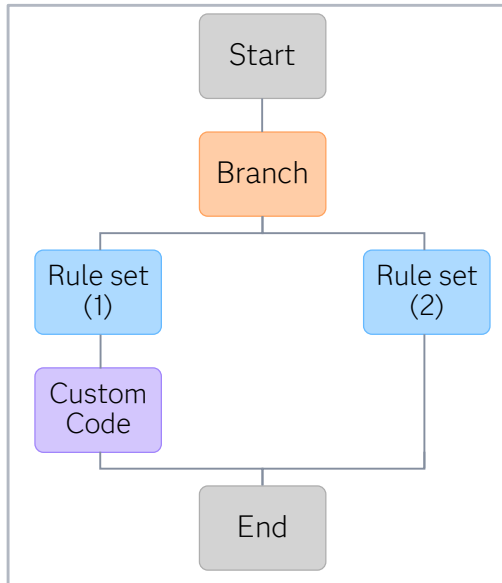


Decisions are a combination of rule sets, analytical models, conditional logic, custom code, and other objects which are used to evaluate data and make real-world business decisions based on SAS decision output.

A business may use Intelligent Decisioning to answer questions like:

- Should a mortgage loan be approved?
- Is a transaction fraudulent?
- Is a machine about to fail?
- Is a medical claim valid?
- Which offer should be sent to a customer?

Decision Flow



Copyright © SAS Institute Inc. All rights reserved.

Decisions can be made up of any combination of object types:

- A/B Testing
- Branch
- Data query
- Decision
- DS2 code file
- Micro Analytic module
- Model
- Rule set
- Treatment group
- ... and more!



When you build a decision, you create a decision flow. You define the steps that you want the decision to perform by adding and configuring nodes that perform different types of processing.

The steps in a decision are called decision objects. Decisions can be made up of any combination of object types, including:

- A/B Testing
- Branch
- Data query
- Decision
- DS2 code file
- Micro Analytic module
- Model
- Rule set
- Treatment group
- ... and more!

Data Scenario: LOAN_APPLICANTS

Objective: decide which customer loans to approve based on application data

LOAN_APPLICANTS Columns: 12 Rows: 2.6 K Size: 364.9 KB Actions

Details Sample data

100 of 2,627 Rows Sample rows: 100

| REQID | CLIENTID | REASON | LOAN | BAD | DEROG | DELINQ | CLAGE |
|-------|---------------|----------|-------|-----|-------|--------|-------------|
| 10000 | 11135157-kwqm | Home ... | 21600 | 0 | 0 | 0 | 626.7713928 |
| 10010 | 11167992-pwpm | Home ... | 26400 | 0 | 1 | 0 | 255.415514 |
| 10020 | 11168688-cxzh | Debt ... | 22000 | 1 | . | 12 | 218 |
| 10040 | 11197158-ikfr | Debt ... | 17400 | 0 | 0 | 0 | 148.9162275 |
| 10050 | 11200784-wuav | Debt ... | 27300 | 0 | 0 | 0 | 144.0934649 |
| 10060 | 11215088-pkyx | Debt ... | 31300 | 0 | 0 | 0 | 230.4130352 |
| 10070 | 11233225-fuhy | Debt ... | 21700 | 0 | 0 | 0 | 154.5525474 |
| 10080 | 11253017-kone | Home ... | 10000 | 1 | 0 | 0 | 171.4333333 |

Copyright © SAS Institute Inc. All rights reserved.



In this workshop, we'll decide which customer loans to approve based on application data from the **LOAN_APPLICANTS** table.

Data Scenario: LOAN_APPLICANTS

| | |
|----------|---|
| REQID | Loan request ID - Unique Identifier |
| CLIENTID | ID for Client who requested loan |
| REASON | Reason for loan request |
| LOAN | Loan amount requested in USD |
| BAD | Defaulted or seriously delinquent account (1 =Yes 0 =No) |
| DEROG | Number of derogatory reports |
| DELINQ | Number of delinquent credit lines |
| CLAGE | Age of oldest credit line in months |
| NINQ | Number of recent credit inquiries |
| DEBINC | Debt to income ratio percentage |

Copyright © SAS Institute Inc. All rights reserved.



LOAN_APPLICANTS contains the loan amount requested by the client plus some information from their credit report. Note that a client could have multiple loan requests.

Data Scenario: LOAN_APPLICANTS

1. Create rule set for initial loan review
 - Objective: Automatically approve or deny loans that meet defined criteria
2. Use a Branch node to separate loans by status
 - "Review", "Approved", or "Denied"
3. Create DS2 code file to process remaining loan applications
 - Objective: Approve or deny remaining loans based on customer credit score, loan amount, and income

Copyright © SAS Institute Inc. All rights reserved.



We'll follow three main steps to build our decision.

First, we'll create a rule set for initial loan review. This will include logic to automatically approve or deny loans that meet the defined criteria.

Next, we'll use a branch node in the final decision to separate loans by status. Loan status could be "Review", "Approved", or "Denied".

Lastly, we'll create a DS2 code file to process the remaining loan applications. This will include logic to approve or deny the remaining loans based on the customer's credit score, loan amount, and/or income.

Rule Sets

Rule sets are collections of rules which specify conditions to be evaluated and actions to be taken if those conditions are satisfied (depending on the rule type).

Rule set types include:

| | |
|------------|--|
| Filtering | IF only rules to select only certain records for processing. |
| Assignment | IF-THEN or IF-THEN-ELSE rules with assignments and/or actions. |
| Common | Shared rules within <i>Assignment</i> type rule sets. |

Copyright © SAS Institute Inc. All rights reserved.



Rule sets are collections of rules which specify conditions to be evaluated and actions to be taken if those conditions are satisfied.

A rule set can be one of three types: **Filtering**, **Assignment**, and **Common**.

- Filtering rule sets only contain **IF** condition expressions; they do not have any action expressions. Filtering rule sets enable you to select only certain records for processing.
- Assignment rule sets are more complex and contain **IF-THEN** or **IF-THEN-ELSE** rules along with **ASSIGN** statements to assign variable values based on the conditions and can include actions for looking up values or exiting the rule set. Assignment rules can also include references to common rule sets and advanced lists.
- Common rule sets enable you to **share the same rules** among different Assignment type rule sets.

Rule Sets

Example: Assignment rule set

The screenshot shows the SAS Rule Sets interface for a rule set named "Initial_Loan_Review (1.0)". The interface includes tabs for "Rule set", "Properties", "Variables", "Scoring", "Versions", and "History". The "Rule set" tab is active, showing a list of rules. The first rule is titled "Automatic Loan Denial or Approval". It contains an "IF" statement with the condition "BAD" and an "THEN" statement with two actions: "ASSIGN STATUS 'DENIED'" and "ASSIGN JUSTIFICATION 'AUTOMATIC DENIAL CRITERIA MET'". Annotations highlight the "ASSIGN" statement and the "IF-THEN" statement.

This example rule set begins by assigning an initial value to the variable *Status*. The assignment statement is followed by an IF-THEN statement which updates *Status* based on other values.

Custom Code

Custom code files can be written and executed in a decision for your specialized processing needs.

Valid custom code file types include:

SAS DS2

**Data Query
(SQL)**

Python

**Custom
context**

Copyright © SAS Institute Inc. All rights reserved.



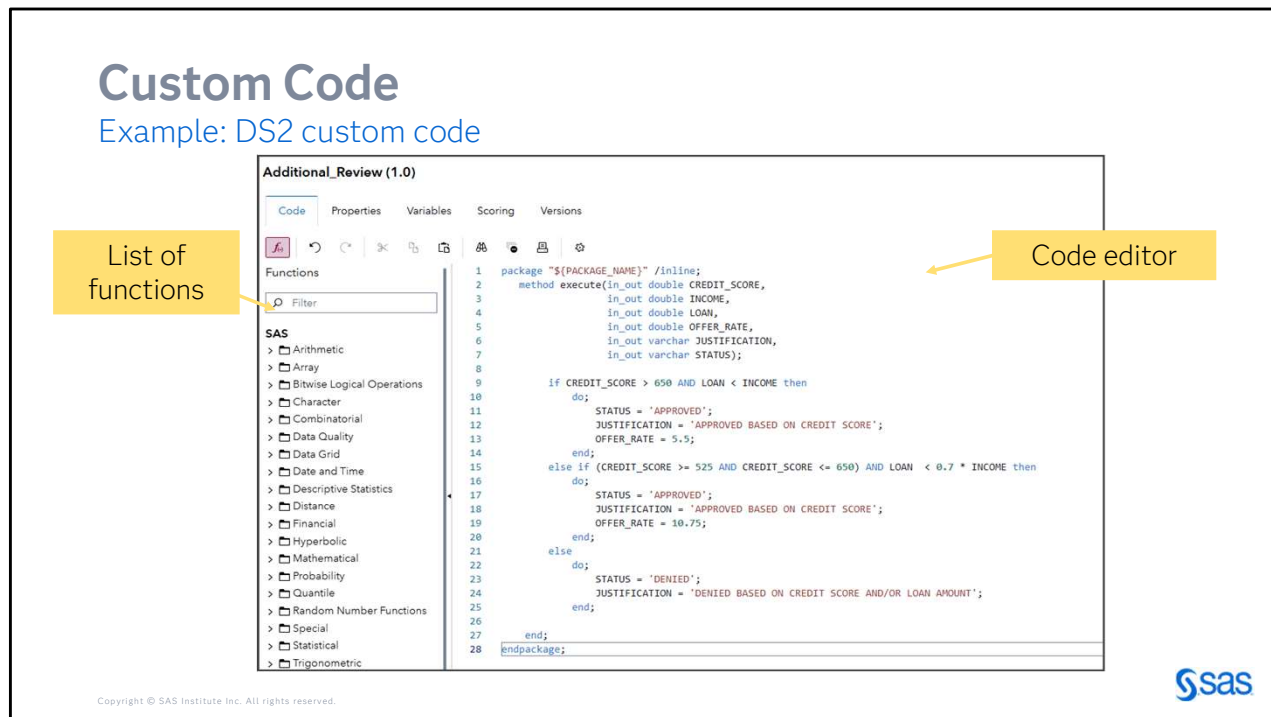
Custom code files can be written and executed in a decision for your specialized processing needs.

Valid custom code file types include:

- SAS DS2
- Data Query (SQL)
- Python
- Custom context

Custom Code

Example: DS2 custom code



The screenshot displays the SAS Custom Code editor interface. On the left, a panel titled 'Additional_Review (1.0)' shows a 'Functions' list with categories like Arithmetic, Array, and Data Quality. A yellow callout box labeled 'List of functions' points to this panel. The main area is the 'Code editor', which contains SAS code for a package named 'execute'. A yellow callout box labeled 'Code editor' points to this area. The code includes variable declarations, conditional logic for loan approval, and package management commands.

```

1 package "${PACKAGE_NAME}" /inline;
2 method execute(in_out double CREDIT_SCORE,
3               in_out double INCOME,
4               in_out double LOAN,
5               in_out double OFFER_RATE,
6               in_out varchar JUSTIFICATION,
7               in_out varchar STATUS);
8
9   if CREDIT_SCORE > 650 AND LOAN < INCOME then
10    do;
11      STATUS = 'APPROVED';
12      JUSTIFICATION = 'APPROVED BASED ON CREDIT SCORE';
13      OFFER_RATE = 5.5;
14    end;
15   else if (CREDIT_SCORE >= 525 AND CREDIT_SCORE <= 650) AND LOAN < 0.7 * INCOME then
16    do;
17      STATUS = 'APPROVED';
18      JUSTIFICATION = 'APPROVED BASED ON CREDIT SCORE';
19      OFFER_RATE = 10.75;
20    end;
21   else
22    do;
23      STATUS = 'DENIED';
24      JUSTIFICATION = 'DENIED BASED ON CREDIT SCORE AND/OR LOAN AMOUNT';
25    end;
26   end;
27 endpackage;
28

```

Copyright © SAS Institute Inc. All rights reserved.

Custom code can be imported into SAS Intelligent Decisioning or written in the custom code editor.

The editor toolbar has many features to assist with developing custom code, including a list of SAS functions that can be inserted into your code.

Hands-on Exercise

sas **innovate** 2025
Copyright © SAS Institute Inc. All rights reserved.



Hands-on exercise is available at <https://github.com/SAS-Innovate-2025/Getting-Started-with-SAS-Intelligent-Decisioning/>.

**Thank you for attending this
workshop!**

sas innovate 2025

Copyright © SAS Institute Inc. All rights reserved.