Risk Score Calculation Methods

When creating a risk program, there are three supported risk score calculation methods:

- Weighted Average
- High Water Mark
- Custom

Weighted Average:

Calculate the mean of a set of risk scores multiplied by pre-defined weights is the typical method for calculating a risk score. For example, imagine five entities, each with a weight of 1, except for one entity that has a weight of 0.5. The lower weight implies that the entity is less important than the others. A set of values might be { 6, 6, 5, 6, and 9 } and the last number is weighted 0.5. For this example, each value is multiplied by its weight:

6 * 1 = 6, 6 * 1 = 6, 5 * 1 = 5, 6 * 1 = 6, 9 * 0.5 = 4.5

6 + 6 + 5 + 6 + 4.5 = 27.5

The total (27.5) is divided by the number of values (5) to find the mean, so the weighted average is 27.5 / 5, or 5.5. Without weighting, the mean of the values would be 6.4. Weighting allows calculations to reflect the relative importance of different entities.

High Water Mark:

In some cases, the highest risk score is most important. The chain is no stronger than its weakest link. The high water mark method makes the risk score equal to the highest risk value in the set. Our example values {6, 6, 5, 6, and 9} would yield a risk score of 9. Because this method does not use weighting, it is most useful while aggregating risk scores for entities of similar importance.

Custom

Risk score calculation can be performed in an external script by specifying the script file in the .properties file, and by creating a script file that implements a specific Java interface in the Groovy language. (Groovy is syntactically similar to Java.)To get started, place the groovy script files in the *config / scripts* folder.

The default script used by the RiskVision system is a helpful reference for users who wants to customize the risk score:

`%AGILIANCE_HOME%\Tomcat\webapps\WEB-INF\classes\scripts\NgErmScriptupdater.groovy`

Treat this file as read-only; never modify files under the WEB-INF folder, because changes can be overwritten without warning.

To use a custom Groovy script for risk score calculations, add the following property to `%AGILIANCE_HOME%\config\agiliance.properties` :

ngerm.riskscore.update.groovy.source=file:%AGILIANCE_HOME%/server/config/NgErmScriptUpdater.groovy

To enable groovy ensure that the following property is configured as false

com.agiliance.web.risk.disableCustomRiskUpdaterGroovy=false

Provide the path to your script file (such as file : `D://main/config/scripts/MyRisk.groovy`). Implement

the following methods:

- customLikelihoodValue(RAUserInput userInput)
- customImpactValue(RAUserInput userInput)
- customResidualLikelihoodValue(RAUserInput userInput)
- customResidualImpactValue(RAUserInput userInput)

You can also override these methods:

- calculateInherentRiskScore
- calculateCurrentRiskScore
- calculateResidualRiskScore

Your custom Groovy script must specify the following package and imports:

```
package com.agiliance.risk;

import java.util.Map;

import com.agiliance.common.ALException;

import com.agiliance.dal.model.LikelihoodDefinition;

import com.agiliance.dal.model.ImpactDefinition;

import com.agiliance.dal.model.RAUserInput;

import com.agiliance.risk.NgErmRiskUpdater;

import com.agiliance.risk.profile.BaseNgErmRiskUpdater;

import com.agiliance.risk.util.RAUtil;

import com.agiliance.common.log.AglLogger
```