

Causality Analytics

for

Analytic Product (name changed)

Client of Predictive Research

Bangalore –560 001

Contents

1. About the document
2. Introduction
3. Driving Technology
4. System Architecture: Componentization
 - a. Inputs
 - b. Pre-processor
 - c. Analytics Processor
 - d. Outputs
 - e. Further Processing/Rendering
5. Java Implementation
6. Resources Required
7. Estimation
8. Conclusions

Casualty Analytics for Analytic Product

1. About the Document

The document is written, by Dr. Sharan Basavaraj Patil on the work of Casualty Analytics Component for Analytic Product. The work was advised by Manager of the Company.

2. Introduction

This document contains the core technology of Casualty Analytics to be embedded into Product as an independent component. The document also contains some sample results of the methods that form the core.

3. Driving Technology

The technological concepts statistical based Correlation Analysis and Regression Analysis form the basis of the technology behind the Casualty Analytics of Analytic Product.

4. System Architecture: Componentization

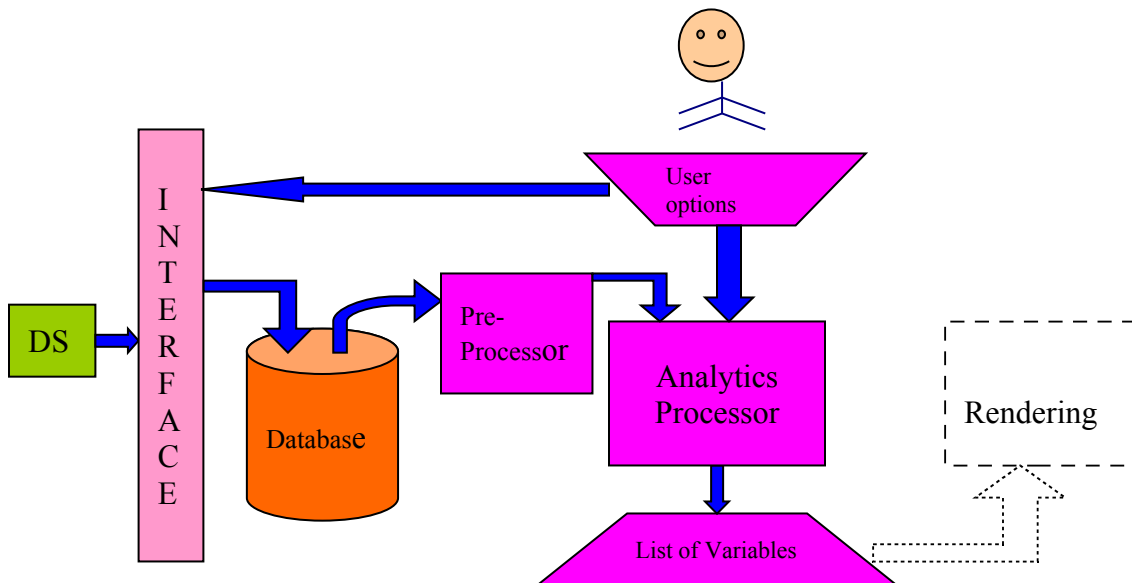


Figure 1: Casualty Analytics Component Architecture

The Casuality Analytics Component Architecture is shown in Figure 1. It presents high-level design of a system and also a data flow.

a. Inputs

The inputs to this Component can be supplied from any data source through the Interfaces (may be tables obtained through SQL Queries). However an important note is these should be tailored based on the user input. Thus, the input data required to give solution to the User Requirement flows to the database

b. Preprocessor

The above input data may contain some of columns, which consists of only one or two elements. Sometimes whole column may contain only zeroes. All these things need to be addressed. In this step, the whole data is preprocessed.

c. Analytics Processor

This is an important part of the whole Architecture. Here the above data is processed in two stages. The procedure is as follows.

1. Is there a necessity to take a two-step process or can be completed in single step. Based on this decision the next two steps are carried out.
2. **Elimination Round:** Apply Correlation Analysis and find out the Correlated Variables. (i.e., which are the variables among given list are related to the Variable under test.
3. **Confirmation Round:** Apply the Regression Analysis to find out which are the variables (among correlated variables) are causes for the Variable under test.
4. Verify the above resultant Variables with domain specific knowledge.

d. Outputs

The outputs **are list of variables, which are effecting** the Variable under Test (or the Variable which User has chosen/opted to study).

e. Further Processing /Rendering

The above obtained output list of variables are further processed to draw the Casuality Diagram.

5. Java Implementation

The method of implementing the above Analytic Processes can be carried out in many ways. Here **some of the simple easy to use and cost effective methods** are mentioned.

i. Using the Java Classes

The class Correlation is available. The information about this class is available at the following site.

<http://openemed.net/javadocs/servers/persistent/javadocs/gov/lanl/PidServer/Correlator.html>

Some other related classes available at

<http://www.sns.gov/APGroup/appProg/xal/xalDoc/gov/sns/tools/correlator/Correlation.html>

The class Regression Model is available. This can be utilized. The information about this class is available at the following site.

<http://openforecast.sourceforge.net/docs/net/sourceforge/openforecast/models/RegressionModel.html>

ii. Using Applets

Some Applets are available at the following sites

<http://wise.cgu.edu/applets/Correl/correl.html>

Some other useful information is available at

<http://home.hia.no/~byrgeb/bb/classes/bbmath/numerical/interpolation/Regression.html>

iii. Using Weka Class Packages

a. Correlation

The details on Correlation classes are available at the following site

<http://weka.sourceforge.net/doc/>

Weka.core consists of correlation method. (**Search in the Class Utils**)

b. Simple Linear Regression Model Classes

<http://weka.sourceforge.net/doc/weka/classifiers/functions/SimpleLinearRegression.html>

6. Resources Deployed

There was a necessity of **only one expert Java Programmer.** (at least some 5 years of experience is required).

7. Estimation

The following was the Time Estimation for the above Resource.

Serial No	Task	Days	Remarks
01.	Familiarization with the concept	01	
02.	Java Class Libraries Usage	02	
03.	Small Tests, Proof Of Concepts, etc.,	02	
04.	Interface/Connectivity Class Designs	02	
05.	Implementation	05	
06.	Testing / Validation	01	
07.	Review & Modifications	02	
08.	Buffer Period	02	
	Total	15(+2)	

8. Conclusions

The Platform Independent Model (Model Driven Architecture) of Casualty Analytics Component which was built for a client is discussed in this document. The simple easy to use and very cost effective methods of Java Implementation are advised to implement the Casualty Analytics.