Oracle Financial Services Retail Portfolio Risk Models and Pooling

User Guide

Release 8.0.0.0.0

December 2014

Contents

List of Figures

1. Introduction

Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) provides the core foundation for delivering the Oracle Financial Services Analytical Applications, an integrated suite of solutions that sit on top of a common account level relational data model and infrastructure components. Oracle Financial Services Analytical Applications enable financial institutions to measure and meet risk-adjusted performance objectives, cultivate a risk management culture through transparency, manage their customers better, improve organization's profitability and lower the costs of compliance and regulation.

All OFSAAI processes, including those related to business, are metadata-driven, thereby providing a high degree of operational and usage flexibility, and a single consistent view of information to all users.

Business Solution Packs (BSP) are pre-packaged and ready to install analytical solutions and are available for specific analytical segments to aid management in their strategic, tactical and operational decision-making.

1.1.Overview of Oracle Financial Services Retail Portfolio Risk Models and Pooling

Under the Capital Adequacy framework of Basel II, banks will for the first time be permitted to group their loans to private individuals and small corporate clients into a Retail Portfolio. As a result, they will be able to calculate the capital requirements for the credit risk of these retail portfolios rather than for the individual accounts. Basel accord has given a high degree of flexibility in the design and implementation of the pool formation process. However, creation of pools can be voluminous and time-consuming. Oracle Financial Services Retail Portfolio Risk Models and Pooling, Release 8.0.0.0.0, referred to as **Retail Pooling** in this document, classifies the retail exposures into segments (pools) using OFSAAI Modeling framework

Abbreviation	Comments
RP	Retail Pooling (Oracle Financial Services Retail Portfolio Risk Models and Pooling)
DL Spec	Download Specification
DI	Data Integrator
PR2	Process Run Rule
DQ	Data Quality
DT	Data Transformation

Table 1: Abbreviations

1.2.Summary

Oracle Financial Services Retail Portfolio Risk Models and Pooling, Release 8.0.0.0.0, product uses modeling techniques available in OFSAAI Modeling framework. The product restricts itself to the following operation:

- Sandbox (Dataset) Creation
- RP Variable Management
- Variable Reduction
  - Correlation
  - Factor Analysis
- Clustering Model for Pool Creation

- Hierarchical Clustering
    - K Means Clustering
- Report Generation
- Pool Stability Report

  OFSAAI Modeling framework provides Model Fitting (Sandbox Infodom) and Model Deployment (Production Infodom). Model Fitting Logic will be deployed in Production Infodom and the Pool Stability report is generated from Production Infodom.

1.3.Approach Followed in the Product

Following are the approaches followed in the product:

- **Sandbox (Dataset) Creation**

  Within the modeling environment (Sandbox environment), data would be extracted or imported from the Production infodom based on the dataset defined there. For clustering, we should have one dataset. In this step, we get the data for all the raw attributes for a particular time period table. Dataset can be created by joining **FCT_RETAIL_EXPOSURE** with **DIM_PRODUCT** table. Ideally one dataset should be created per product, product family, or product class.

- **RP Variable Management**

  For modeling purposes, you need to select the variables required for modeling. You can select and treat these variables in the **Variable Management** screen. You can select variables in the form of Measures, Hierarchy or Business Processors. Also, as pooling cannot be done using character attributes, therefore all attributes have to be converted to numeric values.

  A measure refers to the underlying column value in data and you may consider this as the **direct** value available for modeling .You may select hierarchy for modeling purposes. For modeling purposes, qualitative variables need to be converted to dummy variables and such dummy variables need to be used in Model definition. Dummy variables can be created on a hierarchy.

  Business Processors are used to derive any variable value. You can include such derived variables in model creation. Pooling is very sensitive to extreme values and hence extreme values could be excluded or treated. This is done by capping the extreme values by using outlier detection technique. Missing raw attributes gets imputed by statistically determined value or manually given value. It is recommended to use imputed values only when the missing rate is not exceeding 10-15%.

  Binning is a method of variable discretization or grouping records into '**n**' groups. Continuous variables contain more information than discrete variables. However, discretization could help obtain the set of clusters faster and hence it is easier to implement a cluster solution obtained from discrete variables. For example: Month on Books, Age of the customer, Income, Utilization, Balance, Credit Line, Fees, Payments, Delinquency, and so on, are some examples of variables which are generally treated as discrete and discontinuous.

- **Factor Analysis Model for Variable Reduction**

  Correlation:

  We cannot build the pooling product if there is any co-linearity between the variables used. This can be overcome by computing the co-relation matrix, and if there exists a perfect or almost perfect co-relation between any two variables, one among them needs to be dropped for factor analysis.

  Factor Analysis:

  Factor analysis is a widely used technique of reducing data. Factor analysis is a statistical technique used to explain variability among observed random variables in terms of fewer unobserved random variables called factors. The observed variables are modeled as linear combinations of the factors, plus **error** terms. Factor analysis using principal components method helps in selecting variables having higher explanatory relationships.

Based on Factor Analysis output, the business user may eliminate variables from the dataset which has communalities far from 1. The choice of which variables will be dropped is subjective and is left to you. In addition to this, OFSAAI Modeling Framework also allows you to define and execute Linear or Logistic Regression technique.

Clustering Model for Pool Creation:

There could be various approaches to pool creation. Some could approach the problem by using supervised learning techniques such as Decision Tree methods to split, grow and understand homogeneity in terms of known objectives.

However, Basel mentions that pools of exposures should be homogenous in terms of their risk characteristics (determinants of underlying loss behavior or predicting loss behavior) and therefore instead of an objective method it would be better to use a non objective approach which is the method of natural grouping of data using risk characteristics alone.

For natural grouping of data, clustering is done using two of the prominent techniques. Final clusters are typically arrived at after testing several models and examining their results. The variations could be based on number of clusters, variables and so on.

There are two methods of clustering: Hierarchical and K means. Each one of these methods has its pros and cons given the enormity of the problem. For larger number of variables and bigger sample sizes or presence of continuous variables, K means is a superior method over Hierarchical. Further, Hierarchical method can run into days without generating any dendrogram and hence may become unsolvable. Since, hierarchical method gives a better exploratory view of the clusters formed; it is used only to determine the initial number of clusters that you would start with to build the K means clustering solution. Nevertheless, if hierarchical does not generate any dendrogram at all, then you are left to grow K means method only.

In hierarchical cluster analysis dendrogram graphs are used to visualize how clusters are formed. Since each observation is displayed, dendrograms are impractical when the data set is large. Also dendrograms are too time-consuming for larger data sets. For non-hierarchical cluster algorithms a graph like the dendrogram does not exist.

Hierarchical Clustering:

Choose a distance criterion. Based on that you are shown a dendrogram based on which the number of clusters are decided. A manual iterative process is then used to arrive at the final clusters with the distance criterion being modified in each step. Since hierarchical clustering is a computationally intensive exercise, presence of continuous variables and high sample size can make the problem explode in terms of computational complexity. Therefore, you are free to do either of following:

- Drop continuous variables for faster calculation. This method would be preferred only if the sole purpose of hierarchical clustering is to arrive at the dendrogram.
- Use a random sample drawn from the data. Again this method would be preferred only if the sole purpose of hierarchical clustering is to arrive at the dendrogram.
- Use a binning method to convert continuous variables into discrete variables.

K Means Cluster Analysis:

Number of clusters is a random or manual input or based on the results of hierarchical clustering. This kind of clustering method is also called a k-means model, since the cluster centers are the means of the observations assigned to each cluster when the algorithm is run to complete convergence. Again we will use the Euclidean distance criterion. The cluster centers are based on least-squares estimation. Iteration reduces the least-squares criterion until convergence is achieved.

- **Pool Stability Report**

  Pool Stability report will contain pool level information across all MIS dates since the pool building. It indicates number of exposures, exposure amount and default rate for the pool.

- **Frequency Distribution Report**

  Frequency distribution table for a categorical variable contain frequency count for a given value.

2. Implementing the Product using the OFSAAI Infrastructure

The following terminologies are constantly referred to in this manual:

- **Data Model -** A logical map that represents the inherent properties of the data independent of software, hardware or machine performance considerations. The data model consists of entities (tables) and attributes (columns) and shows data elements grouped into records, as well as the association around those records.
- **Dataset** - It is the simplest of data warehouse schemas. This schema resembles a star diagram. While the center contains one or more fact tables the points (rays) contain the dimension tables (see Figure 1):

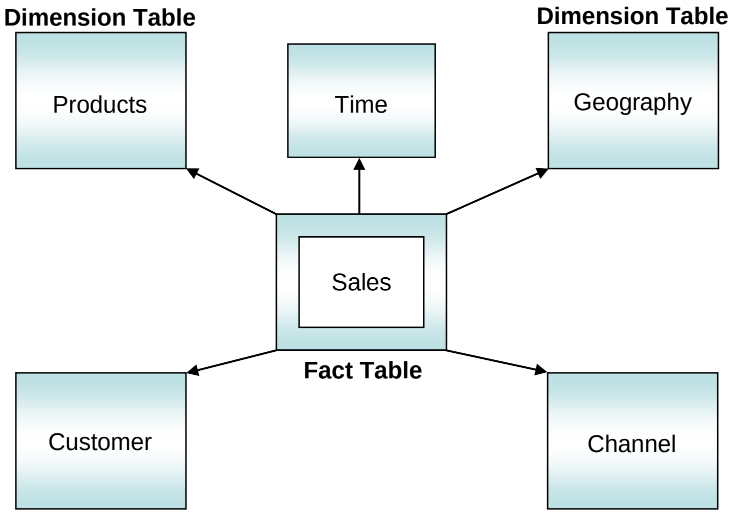

Figure 1: Data Warehouse Schemas

- **Fact Table :**In a star schema, only one join is required to establish the relationship between the FACT table and any one of the dimension tables which optimizes queries as all the information about each level is stored in a row. The set of records resulting from this star join is known as a dataset.
- **Metadata:** is a term used to denote data about data. Business metadata objects are available to in the form of Measures, Business Processors, Hierarchies, Dimensions, Datasets, and Cubes and so on. The commonly used metadata definitions in this manual are Hierarchies, Measures and Business Processors.
- **Hierarchy –** A tree structure across which data is reported is known as a hierarchy. The members that form the hierarchy are attributes of an entity. Thus, a hierarchy is necessarily based upon one or many columns of a table. Hierarchies may be based on either the FACT table or dimensional tables.
- **Measure -** A simple measure represents a quantum of data and is based on a specific attribute (column) of an entity (table). The measure by itself is an aggregation performed on the specific column such as summation, count or a distinct count.

- **Business Processor –** This is a metric resulting from a computation performed on a simple measure. The computation that is performed on the measure often involves the use of statistical, mathematical or database functions.
- **Modelling Framework –** The OFSAAI Modeling Environment performs estimations for a given input variable using historical data. It relies on pre-built statistical applications to build models. The framework stores these applications, so that models can be built easily by business users. The metadata abstraction layer is actively used in the definition of models. Underlying metadata objects such as Measures, Hierarchies, and Datasets are used along with statistical techniques in the definition of models.

2.1.Introduction to Rules

Institutions in the financial sector may require constant monitoring and measurement of risk in order to conform to prevalent regulatory and supervisory standards. Such measurement often entails significant computations and validations with historical data. Data must be transformed to support such measurements and calculations. The data transformation is achieved through a set of defined rules.

The **Rules** option in the Rules Framework Designer provides a framework that facilitates the definition and maintenance of a transformation. The metadata abstraction layer is actively used in the definition of rules where you are permitted to re-classify the attributes in the data warehouse model thus transforming the data. Underlying metadata objects such as Hierarchies that are non-large or non-list, Datasets and Business Processors drive the Rule functionality.

2.1.1. Types of Rules

From a business perspective, Rules can be of 3 types:

- **Type 1:** This type of Rule involves the creation of a subset of records, from a given set of records in the data model, based on certain filters. This process may or may not involve transformations or aggregation or both. Such type 1 rule definitions are achieved through Table-to-Table (T2T) Extract. (Refer to the section *Defining Extracts* in the *Data Integrator User Manual* for more details on T2T Extraction*)*.
- **Type 2:** This type of Rule involves re-classification of records in a table in the data model based on criteria that include complex Group By clauses and Sub Queries within the tables.
- **Type 3:** This type of Rule involves computation of a new value or metric based on a simple measure and updating an identified set of records within the data model with the computed value.

2.1.2. Rule Definition

A rule is defined using existing metadata objects. The various components of a rule definition are:

- **Dataset –** This is a set of tables that are joined together by keys. A dataset must have at least one FACT table. Type 3 rule definitions may be based on datasets that contain more than 1 FACT tables. Type 2 rule definitions must be based on datasets that contain a single FACT table. The values in one or more columns of the FACT tables within a dataset are transformed with a new value.
- **Source –** This component determines the basis on which a record set within the dataset is classified. The classification is driven by a combination of members of one or more hierarchies. A hierarchy is based on a specific column of an underlying table in the data warehouse model. The table on which the hierarchy is defined must be a part of the dataset selected. One or more hierarchies can participate as a source so long as the underlying tables on which they are defined belong to the dataset selected.

- **Target –** This component determines the column in the data warehouse model that will be impacted with an update. It also encapsulates the business logic for the update. The identification of the business logic can vary depending on the type of rule that is being defined. For type 3 rules, the business processors determine the target column that is required to be updated. Only those business processors must be selected that are based on the same measure of a FACT table present in the selected dataset. Further, all the business processors used as a target must have the same aggregation mode. For type 2 rules, the hierarchy determines the target column that is required to be updated. The target column is in the FACT table and has a relationship with the table on which the hierarchy is based. The target hierarchy must not be based on the FACT table.
- **Mapping –** This is an operation that classifies the final record set of the target that is to be updated into multiple sections. It also encapsulates the update logic for each section. The logic for the update can vary depending on the hierarchy member or business processor used. The logic is defined through the selection of members from an intersection of a combination of source members with target members.
- **Node Identifier –** This is a property of a hierarchy member. In a Rule definition the members of a hierarchy that cannot participate in a mapping operation are target members, whose node identifiers identify them to be an 'Others' node, 'Non-Leaf' node or those defined with a range expression. (Refer to the section Defining Business Hierarchies in the Unified Metadata Manager Manual for more details on hierarchy properties). Source members, whose node identifiers identify them to be 'Non-Leaf' nodes, can also not participate in the mapping.

2.2.Introduction to Processes

A set of rules collectively forms a **Process**. A process definition is represented as a **Process Tree**.

The **Process** option in the Rules Framework **Designer** provides a framework that facilitates the definition and maintenance of a process. A hierarchical structure is adopted to facilitate the construction of a process tree. A process tree can have many levels and one or many nodes within each level. Sub-processes are defined at level members and rules form the leaf members of the tree. Through the definition of Process, you are permitted to logically group a collection of rules that pertain to a functional process.

Further, the business may require simulating conditions under different business scenarios and evaluate the resultant calculations with respect to the baseline calculation. Such simulations are done through the construction of Simulation Processes and Simulation Process trees.

Underlying metadata objects such as Rules, T2T Definitions, Non End-to-End Processes and Database Stored Procedures drive the Process functionality.

From a business perspective, processes can be of 2 types:

- **End-to-End Process** – As the name suggests, this process denotes functional completeness. This process is ready for execution.
- **Non End-to-End Process –** This is a sub-process that is a logical collection of rules. It cannot be executed by itself. It must be defined as a sub-process in an end-to-end process to achieve a state ready for execution. A process is defined using existing rule metadata objects.
- **Process Tree -** This is a hierarchical collection of rules that are processed in the natural sequence of the tree. The process tree can have levels and members. Each level constitutes a sub-process. Each member can either be a Type 2 rule or Type 3 rule, an existing non end-to-end process, a Type 1 rule (T2T) or an existing transformation that is defined through Data Integrator. If no predecessor is defined, the process tree is executed in its natural hierarchical sequence as explained in the stated example.

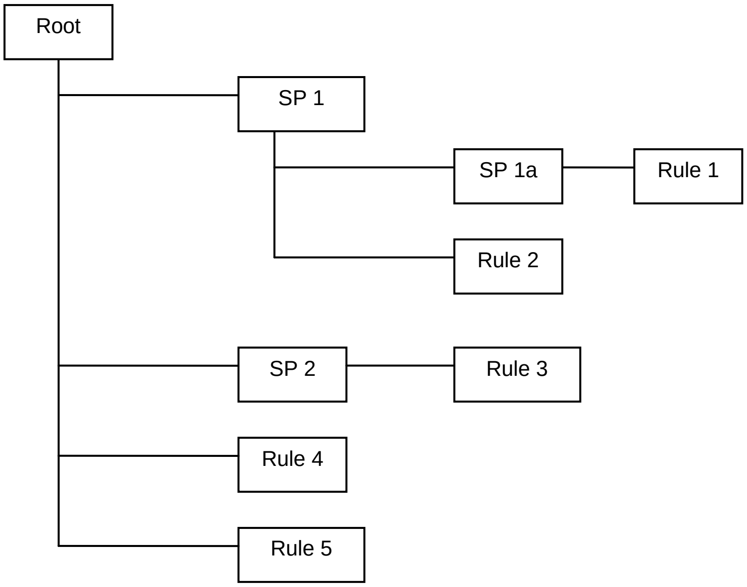

Figure 2: Process Tree

For example: In the above figure, first the sub process SP1 will be executed. The sub process SP1 will be executed in following manner: - **Rule 1 > SP1a > Rule 2> SP1.** The execution sequence will be start with Rule 1, followed by sub-process SP1a, followed by Rule 2 and will end with sub-process SP1:

The Sub Process SP2 will be executed after execution of SP1. SP2 will be executed in following manner: - **Rule 3 > SP2.** The execution sequence will start with Rule 3 followed by sub-process SP2. After execution of sub-process SP2, Rule 4 will be executed, and then finally the Rule 5 will be executed. The Process tree can be built by adding one or more members called **Process Nodes**.

If there are **Predecessor Tasks** associated with any member, the tasks defined as predecessors will precede the execution of that member.

2.2.1. Type of Process Trees

Two types of process trees can be defined.

- **Base Process Tree -** is a hierarchical collection of rules that are processed in the natural sequence of the tree. The rules are sequenced in a manner required by the business condition. The base process tree does not include sub-processes that are created at run time during execution.
- **Simulation Process Tree -** as the name suggests is a tree constructed using a base process tree. It is also a hierarchical collection of rules that are processed in the natural sequence of the tree. It is however different from the base process tree in that it reflects a different business scenario.

The scenarios are built by either substituting an existing process with another or inserting a new process or rules.

2.3.Introduction to Run

In this chapter we will describe how the processes are combined together and defined as '**Run'**. From a business perspective, different 'Runs' of the same set of processes may be required to satisfy different approaches to the underlying data.

The **Run** Framework enables the various Rules defined in the Rules Framework to be combined together (as processes) and executed as different 'Baseline Runs' for different underlying approaches. Different approaches are achieved through process definitions. Further, run level conditions or process level conditions can be specified while defining a 'Run'.

In addition to the baseline runs, simulation runs can be executed through the usage of the different Simulation Processes. Such simulation runs are used to compare the resultant performance / calculations with respect to the baseline runs. This comparison will provide useful insights on the effect of anticipated changes to the business.

2.3.1. Run Definition

A **Run** is a collection of processes that are required to be executed on the database. The various components of a run definition are:

- **Process-** you may select one or many End-to-End processes that need to be executed as part of the Run.
- **Run Condition-** When multiple processes are selected, there is likelihood that the processes may contain rules / T2Ts whose target entities are across multiple datasets. When the selected processes contain Rules, the target entities (hierarchies) which are common across the datasets are made available for defining Run Conditions. When the selected processes contain T2Ts, the hierarchies that are based on the underlying destination tables which are common across the datasets are made available for defining the Run Condition. A Run Condition is defined as a filter on the available hierarchies.
- **Process Condition -** A further level of filter can be applied at the process level. This is achieved through a mapping process.

2.3.2. Types of Runs

Two types of runs can be defined namely **Baseline Runs** and **Simulation Runs**.

- **Baseline Runs -** are those base End-to-End processes that are executed.
- **Simulation Runs -** are those scenario End-to-End processes that are executed. Simulation Runs are compared with the Baseline Runs and therefore the Simulation Processes used during the execution of a simulation run are associated with the base process.

2.4.Building Business Processors for Calculation Blocks

This chapter describes what a **Business Processor** is and explains the process involved in its creation and modification.

The **Business Processor** function allows you to generate values that are functions of base measure values. Using the metadata abstraction of a business processor, power users have the ability to design rule-based transformation to the underlying data within the data warehouse / store. (Refer to the section *defining a Rule* in the *Rules, Process and Run Framework Manual* for more details on the use of business processors*).*

2.4.1. What is a Business Processor?

A **Business Processor** encapsulates business logic for assigning a value to a measure as a function of observed values for other measures.

Let us take an example of risk management in the financial sector that requires calculating the risk weight of an exposure while using the Internal Ratings Based Foundation approach. Risk weight is a function of measures such as Probability of Default (PD), Loss Given Default and Effective Maturity of the exposure in question. The function (risk weight) can vary depending on the various dimensions of the exposure like its customer type, product type and so on. Risk weight, is an example of a business processor.

2.4.2. Why Define a Business Processor?

Measurements that require complex transformations that entail transforming data based on a function of available base measures require business processors. A supervisory requirement necessitates the definition of such complex transformations with available metadata constructs. Business Processors are metadata constructs that are used in the definition of such complex rules. (Refer to the section Accessing Rule in the Rules, Process and Run Framework Manual for more details on the use of business processors).

Business Processors are designed to update a measure with another computed value. When a rule that is defined with a business processor is processed, the newly computed value is updated on the defined target. Let us take the example cited in the above section where, risk weight is the business processor. A business processor is used in a rule definition (Refer to the section defining a Rule in the Rules, Process and Run Framework Manual for more details). In this example, a rule is used to assign a risk weight to an exposure with a certain combination of dimensions.

2.5.Modeling Framework Tools or Techniques used in RP

Oracle Financial Services Retail Portfolio Risk Models and Pooling, Release 8.0.0.0.0, uses modeling features available in the OFSAAI Modeling Framework. Major tools or techniques that are required for Retail Pooling are briefly described in this section. *Please refer OFSAAI Modeling Framework User Manual for usage in detail.*

- **Outlier Detection -** Pooling is very sensitive to Extreme Values and hence extreme values could be excluded or treated. Records having extreme values can be excluded by applying a dataset filter. Extreme values can be treated by capping the extreme values which are beyond a certain bound. This kind of bounds can be determined statistically (using inter-quartile range) or given manually.
- **Missing Value –** Missing value in a variable needs to be impute with suitable values depending on other data values in the variable. Imputation can be done by manually specifying the value with which it needs to be imputed or by using the mean for the variables created from numeric attributes or Mode for variables created from qualitative attributes. If it gets replaced by mean or mode, it is recommended to use outlier treatment before applying missing value. Also, it is recommended that Imputation should only be done when the missing rate does not exceed 10-15%.
- **Binning -** Binning is the method of variable discretization whereby continuous variable can be discredited and each group contains a set of values falling under specified bracket. Binning could be Equi-width, Equi-frequency or manual binning. The number of bins required for each variable can be decided by the business user. For each group created above, you could consider the mean value for that group and call them as bins or the bin values.
- **Correlation -** Correlation technique helps identify the correlated variable. Perfect or almost perfect correlated variables can be identified and the business user can remove either of such variables for factor analysis to effectively run on remaining set of variables.

- **Factor Analysis –** Factor analysis is a statistical technique used to explain variability among observed random variables in terms of fewer unobserved random variables called factors. The observed variables are modeled as linear combinations of the factors, plus *error* terms. From the output of factor analysis, business user can determine the variables that may yield the same result and need not be retained for further techniques.

- **Hierarchical Clustering -** In hierarchical cluster analysis dendrogram graphs are used to visualize how clusters are formed. You can choose a distance criterion. Based on that a dendrogram is shown and based on which the number of clusters are decided upon. Manual iterative process is then used to arrive at the final clusters with the distance criterion being modified with iteration. Since hierarchical method may give a better exploratory view of the clusters formed, it is used only to determine the initial number of clusters that you would start with to build the K means clustering solution.

  Dendrograms are impractical when the data set is large because each observation must be displayed as a leaf they can only be used for a small number of observations. For large numbers of observations hierarchical cluster algorithms can be time consuming. Also, hierarchical clustering is computationally intensive exercise and hence presence of continuous variables and high sample size can make the problem explode in terms of computational complexity. Therefore, you have to ensure that continuous variables are binned prior to its usage in Hierarchical clustering.

- **K Means Cluster Analysis -** Number of clusters is a random or manual input based on the results of hierarchical clustering. In K-Means model, the cluster centers are the means of the observations assigned to each cluster when the algorithm is run to complete convergence. The cluster centers are based on least-squares estimation and the Euclidean distance criterion is used. Iteration reduces the least-squares criterion until convergence is achieved.

- **K Means Cluster and Boundary based Analysis:** This process of clustering uses K-Means Clustering to arrive at an initial cluster and then based on business logic assigns each record to a particular cluster, based on the bounds of the variables. *For more information on K means clustering, refer Annexure C.*

- **CART (GINI TREE) -** *Classification tree* analysis is a term used when the predicted outcome is the class to which the data belongs to. *Regression tree* analysis is a term used when the predicted outcome can be considered a real number. *CART* analysis is a term used to refer to both of the above procedures. GINI is used to grow the decision trees for where dependent variable is binary in nature.

- **CART (Entropy) -** Entropy is used to grow the decision trees where dependent variable can take any value between 0 and 1. Decision tree is a predictive model; that is, a mapping of observations about an item to arrive at conclusions about the item's target value.

3. Understanding Data Extraction

3.1. Introduction

In order to receive input data in a systematic way, we provide the bank with a detailed specification, called a Data Download Specification or a DL Spec. These DL Specs help the bank understand the input requirements of the product and prepare and provide these inputs in proper standards and formats.

3.2. Structure

A DL Spec is an excel file having the following structure:

- Index sheet: This sheet lists out the various entities whose download specifications or DL Specs are included in the file. It also gives the description and purpose of the entities and the corresponding physical table names in which the data gets loaded.
- Glossary sheet: This sheet explains the various headings and terms used for explaining the data requirements in the table structure sheets.
- Table structure sheet: Every DL spec contains one or more table structure sheets. These sheets are named after the corresponding staging tables. This contains the actual table and data elements required as input for the Oracle Financial Services Basel Product. This also includes the name of the expected download file, staging table name, and name, description, data type, and length and so on, of every data element.
- Setup data sheet: This sheet contains a list of master, dimension and system tables that are required for the system to function properly.

The DL spec has been divided into various files based on risk types as follows.

- Retail Pooling
- DLSpecs_Retail_Pooling.xls details the data requirements for retail pools.
- Dimension Tables
- DLSpec_DimTables.xls lists out the data requirements for dimension tables like Customer, Lines of Business, Product and so on.

Annexure A – Definitions

This section defines various terms which are relevant or is used in the user guide. These terms are necessarily generic in nature and are used across various sections of this user guide. Specific definitions, which are used only for handling a particular exposure, are covered in the respective section of this document.

Retail Exposure:

- Exposures to individuals, such as revolving credits and lines of credit (credit cards, overdrafts, and retail facilities secured by financial instruments) as well as personal term loans and leases (installment loans, auto loans and leases, student and educational loans, personal finance, and other exposures with similar characteristics) are generally eligible for retail treatment regardless of exposure size.
- Residential mortgage loans (including first and subsequent liens, term loans and revolving home equity lines of credit) are eligible for retail treatment regardless of exposure size so long as the credit is extended to an individual that is an owner occupier of the property. Loans secured by a single or small number of condominium or co-operative residential housing units in a single building or complex also fall within the scope of the residential mortgage category.
- Loans extended to small businesses and managed as retail exposures are eligible for retail treatment provided the total exposure of the banking group to a small business borrower (on a consolidated basis where applicable) is less than 1 million. Small business loans extended through or guaranteed by an individual are subject to the same exposure threshold. The fact that an exposure is rated individually does not by itself deny the eligibility as a retail exposure.

Borrower risk characteristics:

Socio-Demographic Attributes related to the customer like income, age, gender, educational status, type of job, time at current job, zip code. External Credit Bureau attributes (if available) such as credit history of the exposure like Payment History, Relationship, External Utilization, Performance on those Accounts, and so on.

Transaction risk characteristics:

Exposure characteristics: Basic Attributes of the exposure like Account number, Product name, Product type, Mitigant type, Location, Outstanding amount, Sanctioned Limit, Utilization, payment, spending behavior, age of the account, opening balance, closing balance, delinquency, etc.

Delinquency of exposure characteristics:

Total Delinquency Amount, Pct Delinquency Amount to Total, Max Delinquency, Amount, Number of More equal than 30 Days Delinquency in last 3 Months, and so on.

Factor Analysis:

Factor analysis is a widely used technique of reducing data. Factor analysis is a statistical technique used to explain variability among observed random variables in terms of fewer unobserved random variables called factors.

Classes of Variables:

We need to specify two classes of variables:

- Target variable (Dependent Variable): Default Indictor, Recovery Ratio
- Driver variable(Independent Variable): Input Data forming the cluster product

Hierarchical Clustering:

Hierarchical Clustering gives initial number of clusters based on data values. In hierarchical cluster analysis dendrogram graphs are used to visualize how clusters are formed. As each observation is displayed, dendrograms are impractical when the data set is large.

K Means Clustering:

Number of clusters is a random or manual input or based on the results of hierarchical clustering. This kind of clustering method is also called a *k-means model*, since the cluster centers are the means of the observations assigned to each cluster when the algorithm is run to complete convergence.

Binning:

Binning is the method of variable discretization or grouping into 10 groups where each group contains equal number of records, as far as possible. For each group created above, we could take the mean or the median value for that group and call them as bins or the bin values.

Where p is the probability of the jth incidence in the ith split.

New Accounts

New Accounts are accounts which are new to the portfolio and they do not have a performance history of 1 year on our books.

Annexure B – Frequently Asked Questions

Please refer to the attached Oracle Financial Services Retail Portfolio Risk Models and Pooling Release 8.0.0.0.0 FAQ.

FAQ.doc

Annexure C– K Means Clustering Based On Business Logic

The process of clustering based on business logic assigns each record to a particular cluster, based on the bounds of the variables. Steps 1 and 2 are followed to find out the bounds of each variable for each of the given cluster. Step 3 helps in deciding the cluster id for a given record.

Steps 1 to 3 are together known as a RULE BASED FORMULA.

In certain cases the rule based formula does not return us a unique cluster id, so we then need to use the MINIMUM DISTANCE FORMULA, which is given in Step 4.

1. The first step is to obtain the mean matrix by running a K Means process. The following is an example of such mean matrix which represents clusters in rows and variables in columns:

	V_1	**V_2**	**V3**	**V4**
C_1	15	10	9	57
C_2	5	80	17	40
C_3	45	20	37	55
C_4	40	62	45	70
C_5	12	7	30	20

2. The next step is to calculate bounds for the variable values. Before this is done, each set of variables across all clusters have to be arranged in ascending order. Bounds are then calculated by taking the mean of consecutive values. The process is as follows:

	V_1
C_2	5
C_5	12
C_1	15
C_3	45
C_4	40

The bounds have been calculated as follows for Variable 1:

Less than 8.5 [(5+12)/2]	**C_2**
Between 8.5 and 13.5	**C_5**
Between 13.5 and 30	**C_1**
Between 30 and 42.5	**C_3**
Greater than 42.5	**C_4**

The above mentioned process has to be repeated for all the variables.

Variable 2:

Less than 8.5	**C_5**
Between 8.5 and 15	**C_1**

Between 15 and 41	**C_3**
Between 41 and 71	**C_4**
Greater than 71	**C_2**

Variable 3:

Less than 13	**C_1**
Between 13 and 23.5	**C_2**
Between 23.5 and 33.5	**C_5**
Between 33.5 and 41	**C_3**
Greater than 41	**C_4**

Variable 4:

Less than 30	**C_5**
Between 30 and 47.5	**C_2**
Between 47.5 and 56	**C_3**
Between 56 and 63.5	**C_1**
Greater than 63.5	**C_4**

3. The variables of the new record are put in their respective clusters according to the bounds mentioned above. Let us assume the new record to have the following variable values:

V_1	**V_2**	**V_3**	**V_4**
46	**21**	**3**	**40**

They are put in the respective clusters as follows (based on the bounds for each variable and cluster combination)

V_1	**V_2**	**V_3**	**V_4**
46	**21**	**3**	**40**
C_4	**C_3**	**C_1**	**C_1**

As C_1 is the cluster that occurs for the most number of times, the new record is mapped to C_1.

4. This is an additional step which is required if it is difficult to decide which cluster to map to. This may happen if more than one cluster gets repeated equal number of times or if all of the clusters are unique.

Let us assume that the new record was mapped as under:

V_1	**V_2**	**V_3**	**V_4**
40	**21**	**3**	**40**
C_3	**C_2**	**C_1**	**C_4**

To avoid this and decide upon one cluster, we use the minimum distance formula. The minimum distance formula is as follows:-

$(x_2 – x_1)$ ^2 + $(y_2 – y_1)$ ^2 + ……

Where x1, y1 and so on, represent the variables of the new record and x2, y2 and so on, represent the variables of an existing record. The distances between the new record and each of the clusters have been calculated as follows:-

C_1	**1407**
C_2	**5358**
C_3	**1383**
C_4	**4381**
C_5	**2481**

C_3 is the cluster which has the minimum distance. Therefore, the new record is to be mapped to Cluster 3.

ANNEXURE D: Generating Download Specifications

Data Model for OFS Retail Portfolio Risk Models and Pooling is available on customer request as an ERwin file.

Download Specifications can be extracted from this model. Refer the whitepaper present in OTN for more details.

Oracle Financial Services Retail Portfolio Risk Models and Pooling, Release 8.0.0.0.0
December 2014
Version number 1.0

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/ financial_services/

Due care has been taken to make this Oracle Financial Services Retail Portfolio Risk Models and Pooling User Guide and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this User Manual and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.