Reject Inference

Summary

Reject Inference allows the user to consider the rejected applicants when developing an application scorecard. A new scorecard will be an improvement on a previous one if it can identify the previous accepts who should have been rejected and the previous rejects who should have been accepted. The identification of this swap set is the key to the development of an effective scorecard.

So far in developing the Good/Bad model, only the Accepted applicants' data has been considered. In order to represent a realistic future 'through-the-door' population, Rejects should also be taken into account. As there is no performance for the rejects, it can be inferred from the performance of the Accepts using the Good/Bad model. Reject performance based on that of Accepts, who should inherently perform better than Rejects, is likely to be biased. The Accept/Reject model is therefore used to infer Reject performance more realistically. The process of inferring Reject performance is known as Reject Inference.

The following example illustrates this:

Characteristic Analysis Report: Age of Applicant

Attribute	# Goods	# Bads	G:B Odds	# Accepts	# Rejects	A:R Odds
18	1283	107	12:1	1390	1392	1:1
19 - 21	5694	407	14:1	6101	6130	1:1
22 - 25	6213	777	8:1	6990	2330	3:1
26 - 30	8110	737	11:1	8847	2214	4:1
31 – 40	7265	484	15:1	7749	1548	5:1
41 – 50	6150	308	20:1	6458	1074	6:1
51 – 65	4396	169	26:1	4565	650	7:1
66 +	2799	100	28:1	2899	360	8:1
Total	**41910**	**3089**	**14:1**	**44999**	**15698**	**3:1**

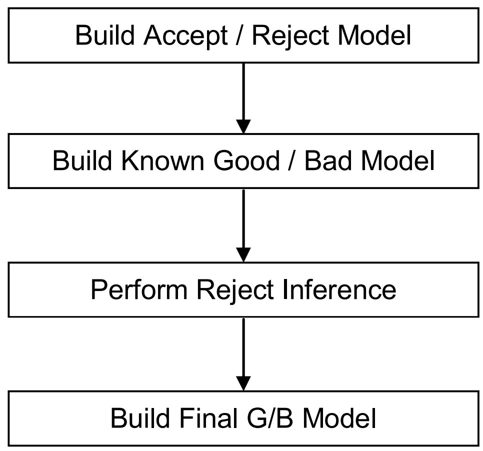
The above report highlights the need to infer a Prob(Good) on the rejected accounts as 18 year old applicants appears to have better G:B Odds than 22-25 year olds – which does not make sense from a risk management perspective. However, if we look at the A:R Odds, we can see that 18 year olds are 3 times more likely to be rejected than the 22-25 year olds. These two age ranges cannot be compared directly without inferring performance on the rejects.

This has effectively shown that only the very best 18 year olds have been accepted whereas a higher risk group of 22-25 year olds was accepted. This would also imply that "Age of

Applicant" might have been used within the current application scorecard / decision making process.

Overview

The following flow diagram describes the recommended high level process involved in completing an application scorecard development:



A more detailed process flow and detail about how to develop a good/bad and an accept/reject model is included in a later section.

In order to perform Reject Inference, the user should ensure that the results of the Good/Bad and Accept/Reject models are available. The results from these models can be used directly, by selecting the relevant model within the reject inference task, or the user can select the relevant probabilities/scores on the data universe. For guidance on the development of these models, see the notes below.

The Reject Inference process creates two reports:

- Reject Inference Line Graph
- Reject Inference Grid

Through these reports the user can adjust and set the Reject performance manually. Once an accepted level of Reject performance has been set the Known Probability of Good for the Accepts and the Inferred Probability of Good for the Rejects is combined to produce a single performance characteristic. This process is referred to as parcelling and the resultant characteristic is the Parcelled (Known and Inferred) Probability of Good for the entire population. Policy Rejects are normally excluded from this process as they will continue to be rejected in the future.

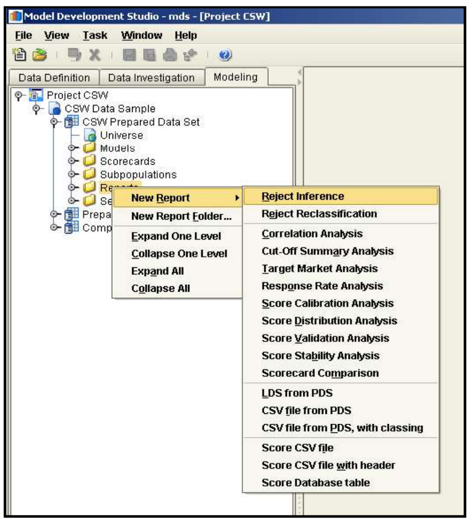
Once the Parcelled characteristic has been generated, the user can use it to develop a model to predict overall performance for the entire population. The results of this model are then used to create the final application scorecard.

The Reject Inference process is essentially a four step *iterative* function. It is suggested that small changes are made to see how this affects the overall distributions, and the process is repeated until the user is happy with the results.

The four steps are:

- Running the Reject Inference reports
- Interacting with the reports to adjust Reject Performance
- Assigning Reject Performance
- Performing validation checks

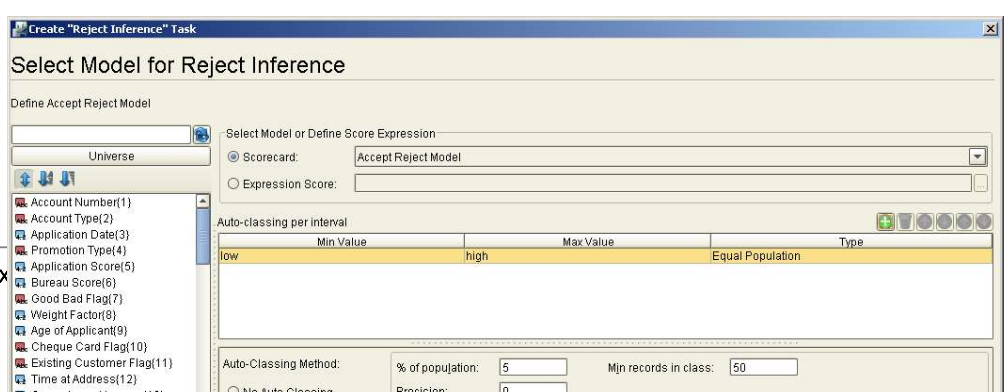
Running the Reject Inference Reports



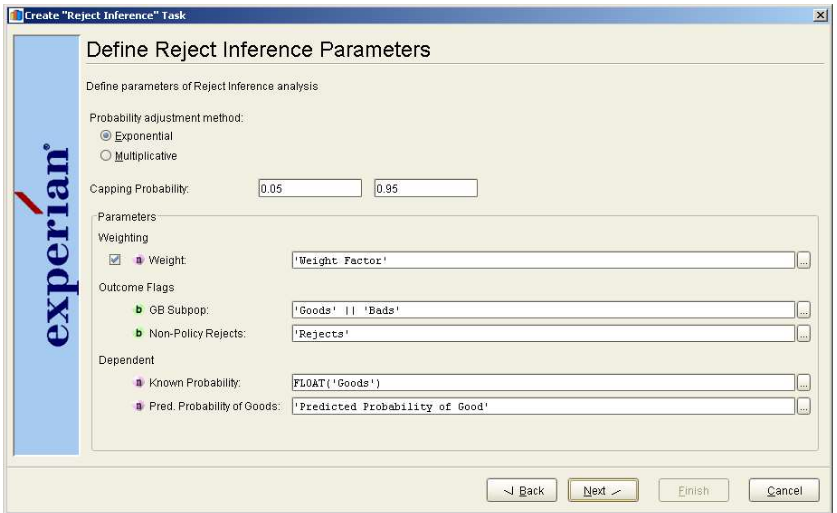
The user can produce a Reject Inference report by right clicking from the reports object from the system view in the modelling tab.

The user is asked to select a sub-population. Typically reject inference is performed on all accepted applications with performance (Goods and Bads) and Non-Policy rejects.

The Select Model for Reject Inference screen asks the user to select the Accept/Reject model/scorecard/score. If an existing application score exists, then this may be used as an alternative to the Accept/Reject model. The user is also asked to define the classing used to create the interpolation points (to be displayed on the interactive graphical report). Typically the 'Equal percentage of the population' option should be used with a value of 5% (this will create 20 equal population bands of accepts, and 20 equal population bands of rejects and is the equivalent of setting the interpoloation points to 20 in earlier versions of the software):



Click **Next** to display the Define Reject Inference Parameters screen:



The user can choose a number of options:

Probability Adjustment Method

- The adjustment method can be either **Exponential** or **Multiplicative**.

Exponential is the most widely used method, and is recommended. With this method, after the manual adjustment has been completed and the parcelling invoked, the line is recalculated using an exponential calculation. The resulting line will lie between the original and adjusted line.

Multiplicative works in a similar way to the above but uses multiplicative equations. The shift of the adjusted line is much less than the exponential method. This is most useful if the user wishes to infer performance and does not wish the inference to move from where they have manually moved it.

Capping Probability

- The Capping Probability is used to adjust any inferred reject probabilities, which fall outside the specified range, to fall within it (defaulted to 0.05 to 0.95).

Weighting

- The appropriate weighting characteristic should be selected.

Outcome Flags

- The definition of Accepts and Non-Policy Rejects. Typically reject inference is performed on all accepted applications with performance (Goods and Bads) and Non-Policy rejects.

Dependent

- The **Known Probability of Good** is the dependent characteristic that was used in the Good/Bad model

The **Predicted Probability of Good** is the characteristic resulting from the Good/Bad model

Interacting with the Reports to Adjust Reject Performance

When the reject inference process has finished, the following reports are displayed:

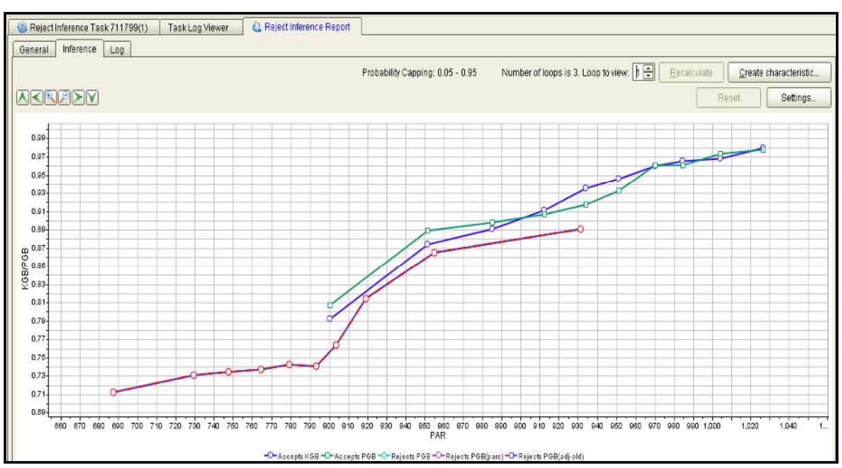
- Reject Inference Line Graph
- Reject Inference Grid

These give graphical and tabular representations of:

- Known Probability of Good for the accepts (Accepts KGB)
- Predicted Probability of Good for the accepts (Accepts PGB)
- Predicted Probability of Good for the rejects (Rejects PGB)

The Line Graph and Grid are interactive and can be used to manually adjust the performance assigned to the rejects.

The following example shows a Reject Inference Line Graph and this will be used to illustrate the values shown:



The graph initially displays three lines. These are the Predicted Probability of Accept by:

- Known Probability of Good characteristic for the Accepts (blue line labelled Accepts KGB)
- Predicted Probability of Good characteristic for the Accepts (green line labelled Accepts PGB)
- Predicted Probability of Good characteristic for the Rejects (turquoise line labelled Rejects PGB, which at first cannot be seen as it is overlaid with red/purple lines)

The closeness of the Accepts KGB and the Accepts PGB points indicates how well the Good/bad model has matched the data.

The Accept/Reject model is judged on the spread of values for the Accepts PGB and the Rejects PGB points.

Typically the trend for the Rejects PGB points is not consistent for observations with a low PAR value. This represents the Good/Bad model not working well for the highly rejected segments of the population.

The Adjusted Predicted Probability of Good for the Rejects line (shown in purple/red) shows the performance expected of the rejects and can be manually adjusted. It is initially overlaid because no adjustments have yet been made.

The values of the Rejects PGB can be edited by dragging the red line graphical points or by amending entries in the Grid.

The following example shows a Reject Inference Line Grid:

Class	PAR	Accepts KGB	Accepts PGB	Rejects PGB	Rejects PGB(parc)	Rejects PGB(adj)	Trans. Val. Adj	Trans. Val. Parc	Rejects PGB(adj-old)
Low : 714.0000	687.3895			0.7126	0.6721	0.6719	1.1734	1.1726	0.6719
714.0000 : 739.0000	729.2840			0.7315	0.6900	0.6898	1.1876	1.1868	0.6898
739.0000 : 753.0000	747.6102			0.7344	0.6985	0.6983	1.1634	1.1623	0.6983
753.0000 : 773.0000	764.3992			0.7370	0.7032	0.7030	1.1547	1.1537	0.7030
773.0000 : 785.0000	779.2481			0.7427	0.7089	0.7086	1.1577	1.1566	0.7086
785.0000 : 798.0000	793.0682			0.7410	0.7172	0.7171	1.1093	1.1087	0.7171
Low : 834.0000	800.5285	0.7922	0.8075						
798.0000 : 810.0000	803.5768			0.7640	0.7261	0.7256	1.1915	1.1891	0.7256
810.0000 : 832.0000	819.1290			0.8147	0.7369	0.7359	1.4963	1.4897	0.7359
834.0000 : 869.0000	851.5316	0.8744	0.8895						
832.0000 : 885.0000	854.8110			0.8653	0.7490	0.7454	2.0306	1.9967	0.7454
869.0000 : 901.0000	885.1131	0.8912	0.8984						
901.0000 : 919.0000	912.1297	0.9110	0.9067						
885.0000 : High	931.3789			0.8910	0.7895	0.7793	2.1609	2.0475	0.7793
919.0000 : 941.0000	933.8404	0.9356	0.9174						
941.0000 : 962.0000	951.0317	0.9462	0.9328						
962.0000 : 975.0000	970.2329	0.9601	0.9608						
975.0000 : 996.0000	984.7189	0.9658	0.9614						
996.0000 : 1012.0000	1004.1137	0.9681	0.9739						
1012.0000 : High	1026.6457	0.9797	0.9778						
Other									

The values contained within in the grid correspond to the lines described above. Also shown are the **transformation value adj**, this is the value the inferred probabilities will be transferred by, and the **transformation value after parcelling**, using the specified adjustment method.

MDS offers the choice of two types of recalculation of the individual PGB for rejects in each class j, based upon adjusted average: 1) Exponential: PGB new = PGBk, k **–**Transformation Value, k=log10(Adjusted Class j PGB) / log10(Initial Class j PGB Average); 2) Multiplicative: PGB new = PGB * k, k=Adjusted Class j PGB / Initial Class j PGB Average.

The user can move the columns in the table by dragging/dropping the column headings within the table. These changes will be saved when the user exits the report or when the report is printed/exported to Excel.

The settings edit button above the graph allows the user to choose the colours displayed on the chart.

Reject performance can be adjusted as many times as required.

After adjustments are completed, the parcelling process is performed, by pressing the recalculate button, which produces a fifth line on the graph. This is the new:

- Predicted Probability of Good characteristic for the Rejects (new purple line labelled Rejects PGB (parc))

These lines are illustrated on the following diagram:

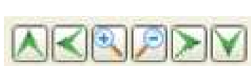
The looping functionality:

available at the top right hand side of the Inference screen allows the user to scroll between iterations.

Iterations are made and stored by selecting the recalculate option, after making adjustments to the Inference graph or table:

(the recalculate option is greyed out if no adjustments have been made). After each adjustment (and recalcualte) the number of loops will increment by 1.

The zoom functionality:



available at the top left hand side of the Inference screen allows the user to zoom in/out on the graph and make detailed adjustments. The user can also move the graph around the screen using the arrows.

Some guidelines that need to be observed when adjusting the rejected inference graph are shown below.

Guideline 1: The rejected population must be inferred with a lower P(Good) than accepts at the same P(Accept) level.

As we are inferring performance on a rejected population, we can not be certain as to how they would have performed. Therefore, we need to be conservative in our estimate, as it would be illogical to state the applicants previously rejected would have performed better than the population accepted.

Guideline 2: For the rejected population with a lower P(Accept) or PAR than the lowest part of the accepted population, the line is adjusted to be lower than a line that would have carried on from the accepted population.

The rejected population that have a very low P(Accept), require treating quite harshly within the reject inference.

Guideline 3: If rejected accounts have a P(Accept) < 0, the manually adjusted P(Good) should approach 0.

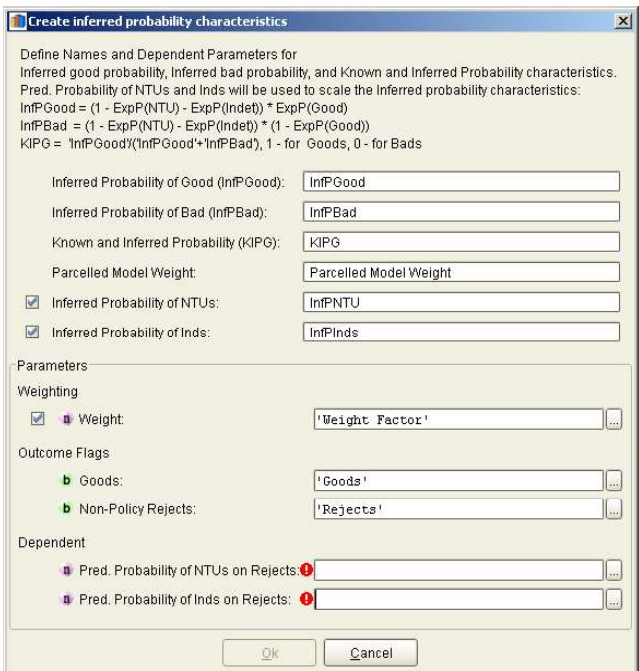
To keep the P(Good) near zero, use the multiplicative method. However, this is not recommended until after the first iteration is validated.

Assigning Reject Performance

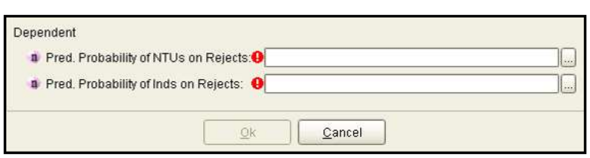
Once the adjustments are complete, it is possible to:

- Specify an Inferred Probability of Good field to store the assigned inferred probabilities
- Specify an Inferred Probability of Bad field to store the assigned inferred probabilities
- Specify a Parcelled Probability of Good field to store the known and assigned inferred probabilities
- Specify a Parcelled Model Weight field to be used in the final model development
- Assign and store a percentage of the rejects as Indeterminate
- Assign and store a percentage of the rejects as Not Taken Up

The user assigns the reject performance by selecting **Create Characteristic**:



The user should first decide on whether a percentage of the rejects should be assigned to be Indeterminate and/or NTU. The user can specify these values in the dependent parameter settings at the bottom of the Create Inferred Probability Characteristics screen (highlighted below).

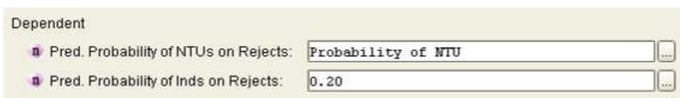


Assigning a Percentage of the Rejects as Indeterminate / NTU

As part of the Reject Inference process it is best practice to assign a proportion of the Rejects to be Indeterminates / NTU. This ensures that the Parcelled results take account of possible Indeterminates / NTU in the Reject population.

Typically a percentage figure is applied, so that the inferred percentage of Indeterminates / NTUs for the Rejects is the same as that for the Accepts. It is also possible to use a model or a combination of these (the edit text box button (…) allows the user full scripting functionality).

The user inputs the desired value (e.g., 20% would be input at 0.20) or model or score field to the dependent parameter settings at the bottom of the Create Inferred Probability Characteristics screen (as shown below):



MDS then uses these two fields (Pred. Probability of NTUs on Rejects (ExpP(NTU)) and the Pred. Probability of Inds on Rejects (ExpP(Indet)) to scale the Expected Probability of Good (ExpP(Good)) to create the Inferred Probability of Good (InfPGood) and the Inferred Probability of Bad (InfPBad) i.e.:

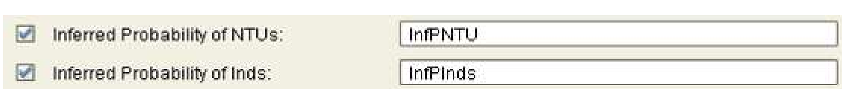
InfPGood = (1 - ExpP(NTU) - ExpP(Indet)) * ExpP(Good)

InfPBad = (1 - ExpP(NTU) - ExpP(Indet)) * (1 - ExpP(Good))

Where the Expected Probability of Good (ExpP(Good)) is the Adjusted Predicted Probability of Good for the Rejects.

The Inferred Probability of Good and Inferred Probability of Bad fields names can be specified on the top half of the Create Inferred Probability Characteristics screen (shown below).

It is recommended that, for each iteration of the reject inference that the user wishes to save, the values used to scale the Expected Probability of Good are also output to the universe. This can be done by selecting the Inferred Probability of NTUs/Inds fields on the top half of the Create Inferred Probability Characteristics screen (shown below). If the user wishes not to do this, they can de-select these fields, and a copy will not be written to the data universe.



Assigning a Known and Inferred Probability of Good (KIPG)

The KIPG field contains the parcelled performance for the accepts and rejects, to be used in the final application model development.

The field contains the Inferred Probability of Good for the Rejects, 1 for the Goods, and 0 for the Bads.

The Known and Inferred Probability of Good field name can be specified on the top half of the Create Inferred Probability Characteristics screen (shown below).

The population also needs to be scaled (to account for the fact that not all rejects are inferred to be good or bad). The Parcelled Model Weight field contains the scaled weight for the rejects (scaled by the number of inferred goods and bads), to be used in the final application model development.

The outcome flag parameters are used to distinguish between observations when MDS makes the above calculations, and can be specified in the middle section of the Create Inferred Probability Characteristics screen (shown below).

Performing Validation Checks

After parcelling has been completed, checks are required on the performance inferred on the rejected accounts. This will show where the inference is in need of alteration. These validation checks should be performed after each iteration of the reject inference.

A simple check to ensure that the inference applied to the rejected accounts is not unrealistic is to compare the overall Good : Bad odds of the known and inferred populations.

This ratio is calculated as follows:

$$\text{Ratio} = \frac{\text{Known Good:Bad Odds (Accepted population)}}{\text{Inferred Good:Bad Odds (Rejected population)}}$$

The ratio will generally be in the range of 3 to 6 depending on the reject rate. For higher reject rates, the ratio may be as low as 2.

It is also recommended that characteristic analysis reports are produced and checked. In the UK, the following reports are typically produced:

Bureau data:

- (Same Person - SP) Number of Non mail-order CAIS 8/9s
- (SP) Time since most recent CAIS 8/9s
- (SP) Total number of CCJs
- (SP) Time since most recent CCJ
- (SP) Total number Searches L3m
- (SP) Total number Searches L6m
- Delphi Score

Application Data:

- Age of Main Applicant
- Main Applicant Residential Status
- Main Applicant Marital Status

These variables are chosen because they have definite known patterns.

The most predictive variables within the Known Good / Bad model should also be verified to ensure that the inference is producing sensible good / bad odds patterns for these variables as well.

The characteristic analysis reports produced will show where the reject inference is currently working correctly and where further alteration may be required. This is demonstrated in the following examples:

(SP) Number of Non mail-order CAIS 8/9s

- A CAIS 8/9 represents an account that has been charged-off or written-off with any company that contributes to the CAIS group (Experian's UK Credit Account Information Sharing). These accounts are associated with applicants that have defaulted on a previous credit agreement. Thus, we can infer that these applicants are very high risk.

Rejected accounts that have any CAIS 8/9 account must look significantly worse than accepted accounts with the same number of CAIS 8/9s.

The two highlighted columns are those of interest:

Fine	<--- Known Applicants			--->	<--- Inferred Applicants			--->	<--- Parcelled Applicants				--->
Interval	# Goods	# Bads	Known G:B Odds	Known G:B index	# Inf Goods	# Inf Bads	Inferred G:B Odds	Inferred G:B Index	# Parc. Goods	# Parc. Bads	Parc. G:B Odds	Parc. G:B Index	Parc. Bad Rate
No trace	263	130	2.02	291B	228	115	1.98	107G	491	245	2.00	135B	32.42
0	88857	14649	6.07	103G	89994	43356	2.08	113G	178851	58005	3.08	114G	24.07
1	1714	585	2.93	201B	10360	9128	1.13	162B	12074	9713	1.24	218B	44.26
2	280	96	2.92	202B	2596	2407	1.08	171B	2876	2503	1.15	236B	46.32
3	77	21	3.67	161B	991	1039	0.95	193B	1068	1060	1.01	269B	49.43
4	10	7	1.43	412B	438	507	0.86	213B	448	514	0.87	311B	53.31
5	15	5	3.00	196B	162	177	0.92	200B	177	182	0.98	277B	50.57
6	17	0	Inf	Inf	105	137	0.76	242B	122	137	0.88	306B	52.46
7	0	1	0.00	Inf	50	53	0.94	196B	50	54	0.92	293B	51.96
8	0	0	Ind	Ind	0	0	Ind	Ind	0	0	Ind	Ind	Ind
9+	0	0	Ind	Ind	65	56	1.15	161B	65	56	1.15	236B	46.61
Others	0	0	Ind	Ind	0	0	Ind	Ind	0	0	Ind	Ind	Ind
TOTAL	91233	15494	5.89	100B	104989	56975	1.84	100B	196222	72469	2.71	100B	26.55

From experience, we know that applicants with a CAIS 8/9 are high risk. This is demonstrated by the “Known G:B Odds”. Accounts with no CAIS 8/9 have odds of 6.07:1 where as accounts with a CAIS 8/9 have odds closer to 3:1.

This report shows the inferred population has been treated logically within the inference process as all odds are below 2:1. Thus, no specific changes to the CAIS 8/9 > 0 population are required.

Delphi Score - Generic bureau score

- A generic score is very useful for assessing the population quickly. As generic scorecards are well tested and robust, they offer a quick assessment of how the G:B odds across the accepted and rejected population are distributed.

Fine	<--- Known Applicants			--->	<--- Inferred Applicants			--->	<--- Parcelled Applicants				--->
Interval	# Goods	# Bads	Known G:B Odds	Known G:B index	# Inf Goods	# Inf Bads	Inferred G:B Odds	Inferred G:B Index	# Parc. Goods	# Parc. Bads	Parc. G:B Odds	Parc. G:B Index	Parc. Bad Rate
LO- 253	197	159	1.24	475B	7124	6629	1.07	171B	7321	6788	1.08	251B	48.04
254- 377	383	284	1.35	437B	6782	6689	1.01	182B	7165	6973	1.03	263B	49.13
378- 455	668	314	2.13	277B	7023	5927	1.19	155B	7691	6241	1.23	220B	44.61
456- 514	667	428	1.56	378B	7653	5359	1.43	129B	8320	5787	1.44	188B	40.78
515- 566	1116	450	2.48	237B	7664	4845	1.58	116B	8780	5295	1.66	163B	37.37
567- 611	1551	512	3.03	194B	7653	4177	1.83	101B	9204	4689	1.96	138B	33.39
612- 648	1860	560	3.32	177B	7626	3500	2.18	118G	9486	4060	2.34	116B	29.66
649- 680	2388	625	3.82	154B	7738	3313	2.34	127G	10126	3938	2.57	105B	27.71
681- 712	2686	748	3.59	164B	7226	3046	2.37	129G	9912	3794	2.61	104B	27.27
713- 745	3515	887	3.96	149B	6547	2485	2.63	143G	10062	3372	2.98	110G	24.67
746- 773	4090	1021	4.01	147B	6219	2385	2.61	142G	10309	3406	3.03	112G	24.36
774- 797	4085	1205	3.39	174B	5699	2180	2.61	142G	9784	3385	2.89	107G	25.25
798- 830	5734	1147	5.00	118B	5038	1727	2.92	158G	10772	2874	3.75	138G	20.62
831- 857	6466	1328	4.87	121B	4119	1433	2.88	156G	10585	2761	3.83	142G	20.13
858- 888	7186	1227	5.86	101B	3481	1127	3.09	168G	10667	2354	4.53	167G	17.62
889- 916	7955	1226	6.49	110G	2650	830	3.19	173G	10605	2056	5.16	191G	15.77
917- 941	8687	1241	7.00	119G	2052	639	3.21	174G	10739	1880	5.71	211G	14.45
942- 965	9641	969	9.95	169G	1536	425	3.61	196G	11177	1394	8.02	296G	10.79
966- 998	10656	680	15.67	266G	844	196	4.31	234G	11500	876	13.13	485G	6.94
999- HI	11702	483	24.23	411G	314	64	4.94	268G	12016	547	21.98	812G	4.27
Others	0	0	Ind	Ind	0	0	Ind	Ind	0	0	Ind	Ind	Ind
TOTAL	91233	15494	5.89	100B	104989	56975	1.84	100B	196222	72469	2.71	100B	26.55

The above example shows that both the known and inferred populations' display increasing odds as the generic score increases. However, the graduation on the known population is much higher than on the inferred population. This would suggest that the middle and lower scoring end of the rejected population require (downward) adjustment within the inference.

Age of Main Applicant

- The application characteristics are included as there are known patterns within applicant data that need to be maintained across the inference.

For age of applicant, it is generally accepted that younger applicants are higher risk than older applicants.

Fine	<--- Known Applicants			--->	<--- Inferred Applicants			--->	<--- Parcelled Applicants				--->
Interval	# Goods	# Bads	Known G:B Odds	Known G:B index	# Inf Goods	# Inf Bads	Inferred G:B Odds	Inferred G:B Index	# Parc. Goods	# Parc. Bads	Parc. G:B Odds	Parc. G:B Index	Parc. Bad Rate
<18	0	0	Ind	Ind	0	0	Ind	Ind	0	0	Ind	Ind	Ind
18- 20	3196	1426	2.24	263B	7452	5469	1.36	135B	10648	6895	1.54	175B	38.74
21- 22	3049	1078	2.83	208B	6562	3938	1.67	111B	9611	5016	1.92	141B	33.75
23- 24	3926	1015	3.87	152B	6598	3792	1.74	106B	10524	4807	2.19	124B	30.83
25	2784	591	4.71	125B	4327	2480	1.75	106B	7111	3071	2.32	117B	29.69
26	3055	573	5.33	110B	4690	2582	1.82	101B	7745	3155	2.45	110B	28.48
27	3232	673	4.80	123B	5020	2777	1.81	102B	8252	3450	2.39	113B	29.12
28	3180	631	5.04	117B	4491	2596	1.73	106B	7671	3227	2.38	114B	29.19
29	3352	578	5.82	101B	4539	2568	1.77	104B	7891	3144	2.51	108B	28.04
30	3531	562	6.28	107G	4319	2250	1.92	104G	7850	2812	2.79	103G	26.02
31	2842	584	4.87	121B	4575	2425	1.89	102G	7417	3009	2.47	110B	28.45
32	3402	571	5.96	101G	4049	2097	1.93	105G	7451	2668	2.79	103G	26.01
33- 34	6529	1061	6.15	105G	7325	3667	2.00	108G	13854	4728	2.93	108G	25.09
35- 36	6025	849	7.10	121G	6753	3547	1.90	103G	12778	4396	2.91	107G	25.22
37- 38	5609	857	6.54	111G	6075	3191	1.90	103G	11684	4048	2.89	107G	25.35
39- 40	4607	676	6.82	116G	4537	2530	1.79	103B	9144	3206	2.85	105G	25.56
41- 42	4781	648	7.38	125G	4207	2094	2.01	109G	8988	2742	3.28	121G	22.99
43- 45	5964	805	7.41	126G	5225	2515	2.08	113G	11189	3320	3.37	124G	22.50
46- 50	9241	1043	8.86	150G	6215	2898	2.14	116G	15456	3941	3.92	145G	19.95
51- 55	6568	639	10.28	175G	4254	1887	2.25	122G	10822	2526	4.28	158G	18.55
56- HI	6360	636	10.00	170G	3776	1672	2.26	123G	10136	2308	4.39	162G	18.14
Others	0	0	Ind	Ind	0	0	Ind	Ind	0	0	Ind	Ind	Ind
TOTAL	91233	15494	5.89	100B	104989	56975	1.84	100B	196222	72469	2.71	100B	26.55

There is a noticeable increase in odds as the age increases for the known population. However, there is little graduation for the inferred population. It should be checked that this characteristic is being catered for appropriately (within the Accept/Reject and Good/Bad models) before the inference is completed.

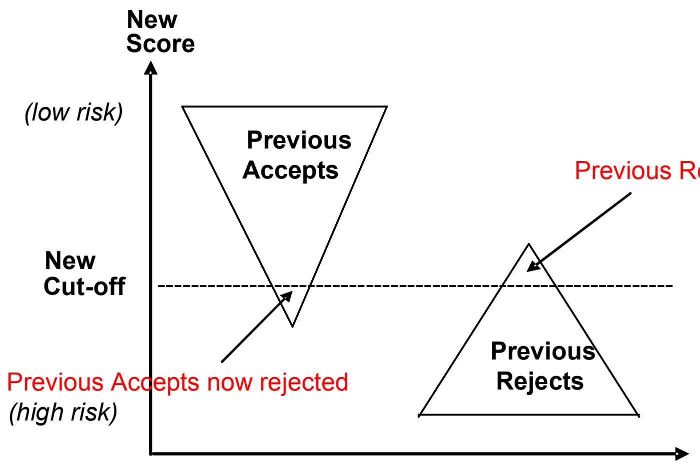
Main Applicant Residential Status

- Through experience, it is known that renters are high risk whilst homeowners are a better risk. This pattern requires replicating within the inferred population to ensure the characteristic retains predictive power.

Main Applicant Marital Status

- Again, as with residential status, experience tells us that married applicants are a better risk than single applicants. This also requires checking before and after reject inference.

The acid test of the validity of the reject inference is to assess the likely size of the swap sets that would be generated with the same overall reject rate. A swap set is the cross over of previous accepts that would be rejected with a new scorecard and previous rejects that would now be accepted using the new scorecard – as shown below.



In order to do this, a “Test” model is developed. The aim of the "Test" model is threefold:

1. to show that if the current reject rate is maintained, we are able to reject a group of high risk previously accepted cases and "swap" them with a group of lower risk previously rejected cases;
2. to ensure that the "% improvement" that the Final model will achieve is believable i.e. in the region of ~15-30% (note: "% improvement" is the % reduction in Bads accepted for the same number of Goods accepted).
3. to ensure that the percentage of previous rejects to be accepted is not too high i.e. generally less than 25%.

The "Test" model is produced in much the same way as the Known Good / Bad Model – this time ensuring the Dependant Variable is known and inferred P(Good).

The coarse classing from the Known Good / Bad model is used, along with any generic bureau scores, if available.

The final step of the model is chosen, as the model is only going to be used to check the reject inference, and is not the final solution.

The “Test” model is used to produce a score distribution. This is shown in the example below:

Previous Accepts now rejected

New Accepts

New Rejects

Final Score	<--- Known Applicants # Goods	# Bads	Known G:B Odds	---> Known G:B index	<--- Inferred Applicants # Inf Goods	# Inf Bads	Inferred G:B Odds	---> Inferred G:B Index	<--- Parcelled Applicants # Parc. Goods	# Parc. Bads	Parc. G:B Odds	Parc. G:B Index	---> Parc. Bad Rate
318.00	191	285	0.67	879B	4857	25002	0.19	640B	5048	25287	0.20	1034B	83.24
421.00	1291	1067	1.21	487B	10923	15104	0.72	172B	12214	16171	0.76	273B	56.13
485.00	2295	1391	1.65	357B	13317	10306	1.29	104G	15612	11697	1.33	154B	42.17
532.00	4066	1936	2.10	280B	13105	7605	1.72	139G	17171	9541	1.80	114B	34.79
576.00	6066	2121	2.86	206B	12424	5643	2.20	177G	18490	7764	2.38	115G	29.09
619.00	8601	2350	3.66	161B	11441	3937	2.91	234G	20042	6226	3.19	155G	23.22
664.00	11967	2325	5.15	114B	9254	2485	3.72	299G	21221	4810	4.41	214G	18.04
715.00	15693	2192	7.16	122G	7261	1272	5.71	459G	22954	3464	6.63	321G	12.74
780.00	19529	981	19.19	326G	4593	576	7.98	641G	24122	1557	15.49	752G	8.44
High	21366	816	26.18	445G	2438	124	19.67	1582G	23804	940	25.32	1229G	3.96
TOTAL	91065	15464	5.89	100B	89613	72054	1.24	100B	180678	87518	2.06	100B	32.12

Previous Rejects now accepted

This score distribution illustrates all 3 of the points described above:

- At a cut-off at 576 (roughly the same reject rate as currently used for this population), the previously accepted (known) population with G:B odds of 2.86:1 and lower would be rejected, and the previously rejected (inferred) population with G:B odds of 2.91:1 and higher would be accepted.
- The percentage of bad accounts, using both the current and new scorecards, when accepting the same number of good accounts is:

	Current Scorecard (Known)	Proposed Scorecard (Known & Inferred)
Number of Good Accounts Accepted	91,065	92,101 (cut-off at 619)
Number of Bad Accounts Accepted	15,464	10,771

The "% improvement" for this Test model is therefore **30%.**

- And finally, the previous rejects to be accepted at a cut-off of 619 include 23,546 goods and 4,457 bads i.e. accounting for 17.3% of the previous (non-policy) rejects.

Notes on Building an Accept Reject Model

The Accept / Reject model is used to modify the predicted Prob (Good) from the Known Good / Bad model before applying it to the rejects, as follows:

Prob (Accept)	Prob (Good) for Rejects
0	<< Prob (Good) for Accepts (Known Good / Bad)
	< Prob (Good) for Accepts (Known Good / Bad)
1	~ Prob (Good) for Accepts (Known Good / Bad)

This model predicts the "acceptability" (or Probability of Accept) of an applicant based on what happened on the previous scoring system.

Unless the previous application scorecard cut-off was applied rigorously, an Accept / Reject model should be built on each segment of the overall portfolio that requires a "master" scorecard. If the previous cut-off was applied rigorously, then the previous applicant score can be used to provide Prob (Accept) for the reject inference process.

Note: Characteristics may predict in a different way for the A / R model versus the Known G/B model.

Accept / Reject Models are built to discriminate between accepted and rejected cases, although "policy" rejects are generally excluded from the A/R models. This is because the A/R models are used to infer G/B performance on the rejects, but the "policy" rejects are - by definition - excluded from the final G/B models and therefore require no performance to be inferred.

The following population should therefore be used within the Accept / Reject Model:

Type of Account	Definition	Include in AR Model?	P(Accept)	Notes
Good	Based on client good / bad definition	Yes	1	
Bad	Based on client good / bad definition	Yes	1	
Indeterminate	Based on client good / bad definition	Yes	1	
Not Taken Up (NTU)	Applicants that are accepted for the credit product but do not finalise the agreement	Yes	1	The definition of NTU will change depending on the population under analysis. The NTU population may consist of a random sample of applicants that have applied and not signed the credit agreement. It will generally have a sample bias towards the better quality end of the sample. As this population is low risk, they may have obtained another similar product from a competitor at a better rate. For Telecom clients, higher risk cases are sometimes asked for a deposit. When the deposit is not supplied, these cases should be treated as "non-policy rejects" - see below.
Non-Policy Reject	Applicant declined based on the score and/or underwriter decision around scorecard cut-off.	Yes	0	
Policy Reject	Applicant declined based on satisfying a policy rule irrespective of the score. e.g., Under 18s, 3+ CCJ's.	No	-	"Policy" rejects are defined as those rejects that are declined through strict rules that will be in place after the scorecard development is completed. Applicants that were rejected through policy rules at the time of sampling, but by rules that have subsequently been rescinded, may be classed as non-policy rejects, or as exclusions to be analysed at the end of the development. Non-policy rejects consist of all other rejects – score declines and/or under-writer declines.
Exclusions	Applicant that are to be excluded from the scorecard development. e.g., Staff accounts, suspected frauds.	No	-	Exclusions are applicants to be removed from the building of the application scorecard models, although they may be scored at the end of the project.

No Development/Validation sampling is required before the Prob (Accept) model build - the total sample should always be used.

It is felt that some over-modelling can be performed for the Accept / Reject model. There are two main measures of over-modelling that are accepted standards:

- Mallows Statistic C(p) > Number of Steps in Model
- Probability F < 0.1000

Traditionally, Mallows Statistic > Number of Steps has been used to decide whether to include a step or not. It has been shown that this measure is still valid, but for accept / reject models it is felt that the Probability F measure is preferable.

For the Accept / Reject model the following rule for using a given step has been defined:

All variables must have Probability F < 0.1000

As with all Experian models, the normal 'reality checks' should be applied to the Prob (Accept) model to ensure that all the co-efficients are explainable.

Notes on Building a Good Bad Model

The Known Good / Bad model is built on accepted good and bad applicants only.

The primary objective of the Known Good / Bad model is to discriminate between good and bad applicants and assign a predicted probability of good to these applicants. This model is then applied to the rejected cases to calculate a predicted probability of good.

As with the Accept / Reject modelling sample, the Known Good / Bad model should be developed on the full population, not just the development sample.

Again, over-modelling within the Good / Bad model is acceptable along the same lines as the Accept / Reject model.

Known Good / Bad Models are built to discriminate between good and bad accounts.

The following population should be used within the Known Good / Bad Model:

Type of Account	Definition	Include in AR Model?	P(Good)	Notes
Good	Based on client good / bad definition	Yes	1	
Bad	Based on client good / bad definition	Yes	0	
Indeterminate	Based on client good / bad definition	No	-	
Not Taken Up (NTU)	Applicants that are accepted for the credit product but do not finalise the agreement	No	-	
Non-Policy Reject	Applicant declined based on the score and underwriter decision around scorecard cut-off.	No	-	
Policy Reject	Applicant declined based on satisfying a policy rule irrespective of the score. e.g. Under 18s, 3+ CCJ's.	No	-	
Exclusions	Applicant that are to be excluded from the scorecard development. e.g. Staff accounts, suspected frauds.	No	-	

An extra step in the Known Good / Bad model solution is to build a "High-Risk Niche" on the lower scoring portion of the Known Good / Bad population. The high-risk niche generally increases the discrimination on the lower scoring (higher risk) portion of the population by ~5-10%.

The high-risk niche is built by modelling the Prob (Good) residuals on the lowest scoring 25-35 % of the population used within the Known Good / Bad master model.

This is generally where the majority of the rejects are distributed and therefore the resulting High Risk Niche model should be significantly more predictive when applied to the overall reject population.

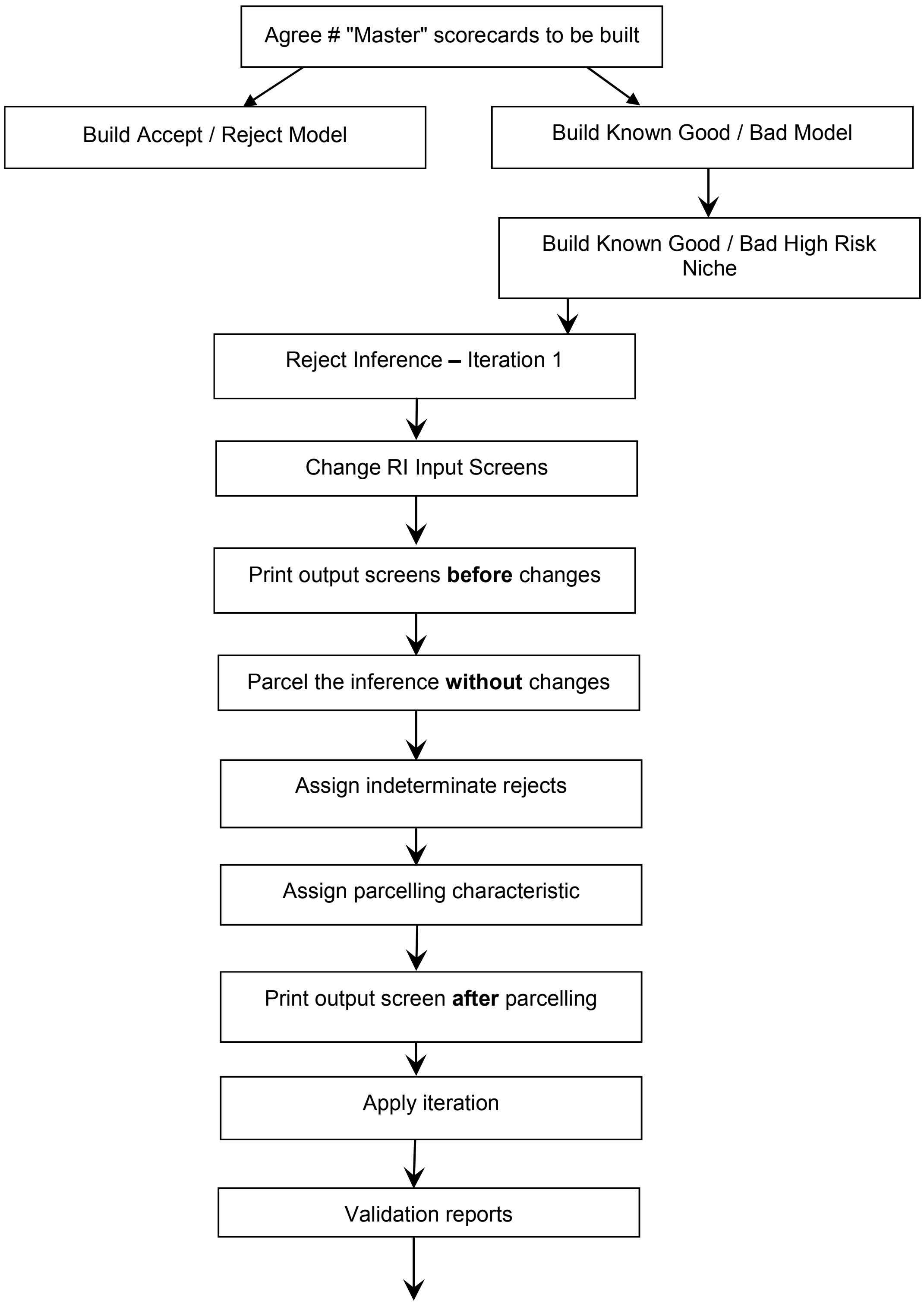
Example:

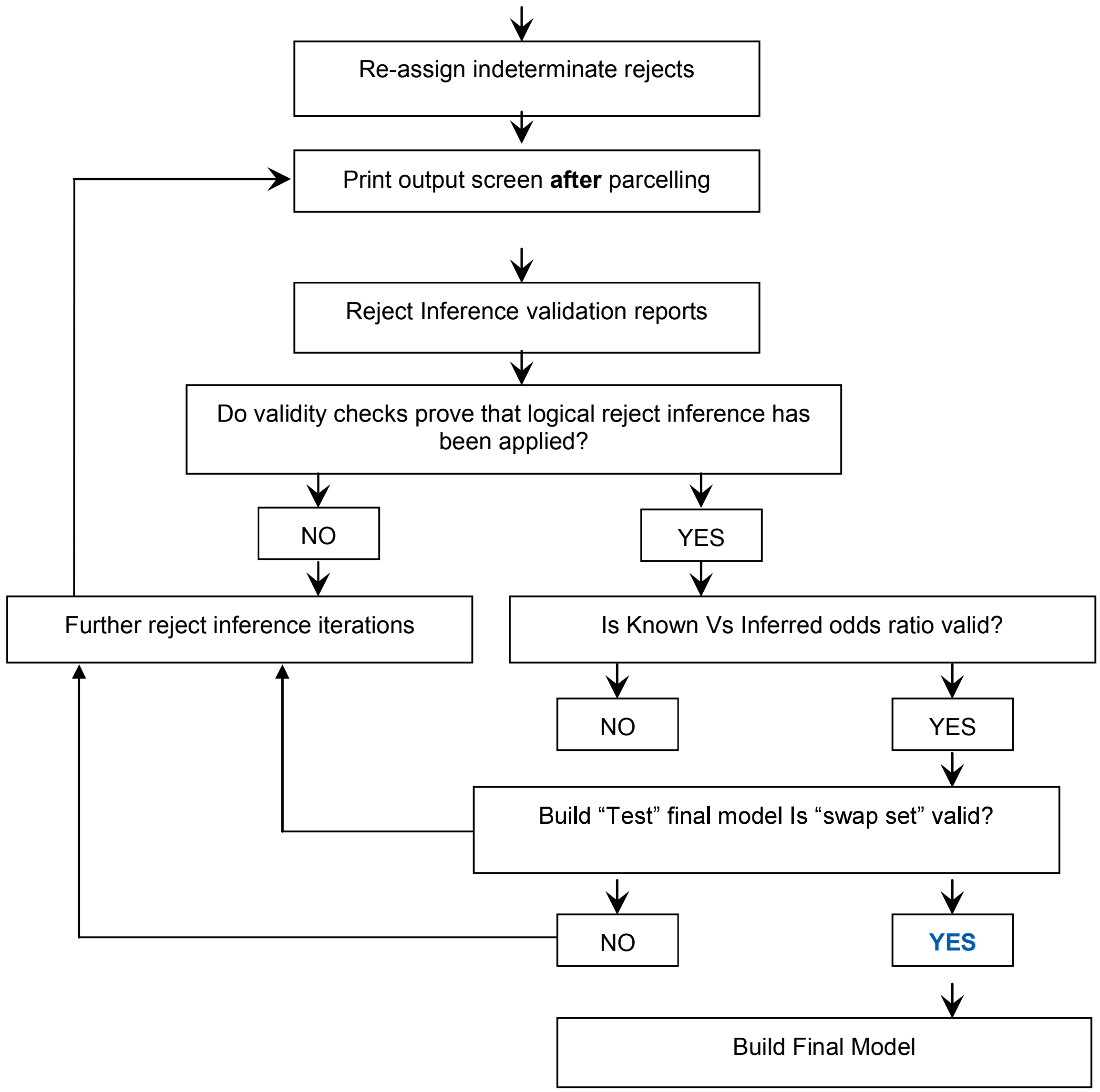
The Known Good / Bad model score distribution for an accepted and rejected population is shown in the following diagram.

Final Score	<--- Known Applicants # Known Goods	# Known Bads	Known G:B Odds	Known G/B index	---> # Accepts	% Accepts	<--- Inferred Applicants # Inf Apps	---> % Inf Apps	Reject Rate	<--- Total # Apps	---> % Apps
643.00	2702	2651	1.02	578B	5718	5.16	68708	42.50	92.32	74852	26.39
694.00	3356	1913	1.75	336B	5618	5.07	22268	13.77	79.85	28284	9.97
727.00	3820	1512	2.53	233B	5635	5.08	16080	9.95	74.05	22166	7.82
754.00	3971	1400	2.84	208B	5687	5.13	11861	7.34	67.59	18031	6.36
778.00	4091	1248	3.28	180B	5630	5.08	9314	5.76	62.33	15391	5.43
797.00	4423	964	4.59	128B	5680	5.12	6522	4.03	53.45	12657	4.46
815.00	4412	863	5.11	115B	5557	5.01	5587	3.46	50.13	11643	4.11
833.00	4508	839	5.37	110B	5642	5.09	4957	3.07	46.77	11286	3.98
850.00	4507	709	6.36	108G	5476	4.94	3910	2.42	41.66	9901	3.49
865.00	4726	657	7.19	122G	5608	5.06	2652	1.64	32.11	8812	3.11
880.00	4792	486	9.86	167G	5464	4.93	2130	1.32	28.05	8226	2.90
896.00	4836	504	9.60	163G	5500	4.96	1838	1.14	25.05	7917	2.79
912.00	5100	402	12.69	215G	5665	5.11	1302	0.81	18.69	7653	2.70
927.00	4721	305	15.48	263G	5175	4.67	1188	0.73	18.67	6968	2.46
943.00	5065	285	17.77	302G	5489	4.95	1318	0.82	19.36	7416	2.62
962.00	5133	253	20.29	345G	5573	5.03	719	0.44	11.43	6935	2.45
982.00	5165	187	27.62	469G	5478	4.94	632	0.39	10.34	6728	2.37
1004.00	5258	145	36.26	616G	5488	4.95	340	0.21	5.83	6443	2.27
1034.00	5177	94	55.07	935G	5346	4.82	271	0.17	4.82	6226	2.20
High	5302	47	112.81	1916G	5404	4.88	70	0.04	1.28	6050	2.13
TOTAL	91065	15464	5.89	100B	110833		161667		59.33	283585	

This shows clearly that the vast majority of the inferred applicants (i.e. the rejects) score in the region where the High Risk Niche model would be developed.

Reject Inference Process Flow




Re-assign indeterminate rejects
Print output screen **after** parcelling
Reject Inference validation reports
Do validity checks prove that logical reject inference has been applied?
NO
YES
Further reject inference iterations
Is Known Vs Inferred odds ratio valid?
NO
YES
Build "Test" final model Is "swap set" valid?
NO
YES
Build Final Model

Reject Reclassification

The Experian North-America analytical team uses a different approach to Reject Inference: *augmentation*, which in the MDS menu is called **Reject Reclassification**. In this method the worst part of the Rejects sub-population, according to the score obtained either by some previously trained Accept/Reject model, or by some generic score, is assigned the Bad value of the Good/Bad flag (usually the 0 value); so the final model uses as the dependent variable the combined Goods and Bads and the worst part of the Rejects, also marked as Bads. In order to preserve population proportions, the records used for the final model should be re-weighted correspondingly, so the algorithm produces new sets of sample weights.

The Reject Reclassification algorithm takes the following parameters:

- Percentage of rejects to reclassify as bad
- Desired ratio of *reclassified bad* to *known bad* (R2K)
- Score obtained by applying accept/reject model (or some generic score, or prior scorecard score)
- Set of characteristics to test reclassification effects

The Score and the value of the percentage of rejects to reclassify as bad are used to come up with the value of a Score threshold, below which the Reject records will be reclassified as bad.

The idea of the approach is as follows: The reject accounts below a certain threshold based on the Score (e.g., Accept/Reject model and/or seriously derogatory information) are reclassified as bad. Then the reclassified reject accounts are weighted so that the ratio of true bad accounts (bad and loss from the accept population) to reclassified accounts is around the ratio of reclassified bad to known bad (default value 30/70) before application sample weights. This purely heuristic ratio can be corrected later on if it is not supported by analysis of the statistics collected for the selected characteristics, and the newly obtained Good/Bad flag (BG1).

Characteristics for validation are selected based on their known, stable, and predictable behaviour in the past.

The necessary statistics are provided in the “Score Cut-off Validation on Rejects” tab and “Weight Validation” tab of the Reject Reclassification Report.

The Reject Reclassification procedure returns the following results:

- New Good/Bad flag, containing 0 values for some of rejects (BG1)
- New sample weights (RATDWGT)
- New good/bad weights (BGEQWGT)

The results can be obtained in the form of a script by clicking the “Create Derived Characteristic” button in the Reject Reclassification Report.

Here is a detailed description of the logic of the Reject Reclassification algorithm, with a numerical example:

Notation: BGRIL/Missing classification – column defined as:

0 - Bad

1 - Good

2 - Indeterminate/NTU

3 - Rejected

4 - Loss (write-offs),

TDWGT – initial sample weights (usually obtained by stratified sampling)

RASCORE – accept/reject score

Algorithm:

```
/* calculates record level BG1, RATDWGT, BGEQWGT */
/* first define actual BG - good/bad flag */

If BGRIL = 0 Or BGRIL = 4 Then
      BG = 0              /* i.e.  # Bads + #Losses = Tot. 
Known Bad */
Else
      If BGRIL = 1 Then
BG = 1
      Else         /* Indetermined/NTU or Reject */
            BG = '.'
      End If
End If
/* define actual accept/reject flag */
If  BGRIL = 3 Then RA = 0 Else RA = 1

If BG = 0 Or BG = 1 Then   /* True Good/Bad flag has a 0/1 
value */
      BG1 = BG
      RATDWGT = TDWGT
Else  /* BG= '.', rejects */

/* Threshold below is determined by the RASCORE and the 
percentage of rejects to reclassify as bad */
      If RA = 0 AND RASCORE < Threshold Then
```



```
BG1 = 0  /* All low score rejects are marked as bad, need
to be re-weighted */
            RATDWGT = TDWGT * R2K * (Tot. Known Bad /
Reclassified Bad)
      Else
            BG1 = BG
RATDWGT = TDWGT
      End If  /* Reclassified Bad =  # of rejects having
score less than Threshold */
End If

If BG1 = 0 Then
     BGEQWGT = RATDWGT * (# Good / Tot. Known Bad) / (1 +
R2K)
Else
     BGEQWGT = RATDWGT
End If
```

BGEQWGT weights are usually used for linear regression performed later on for validation purposes.

BG1 includes 0 values for some percentage of the rejects, and later on is used to build a good/bad model for scoring.

Numerical example

Pop.	Sample	BGRIL	RA	BG	TDWGT	BG1	RATDWT	BGEQWGT
100	100	0 B	1	0	1	0	1	0.58
1000	100	1 G	1	1	10	1	10	10
200	40	2 R	0	.	5	. or 0	22.96	13.39
30	30	3 I	1	.	1	.	1	1
20	20	4 L	1	0	1	0	1	0.58

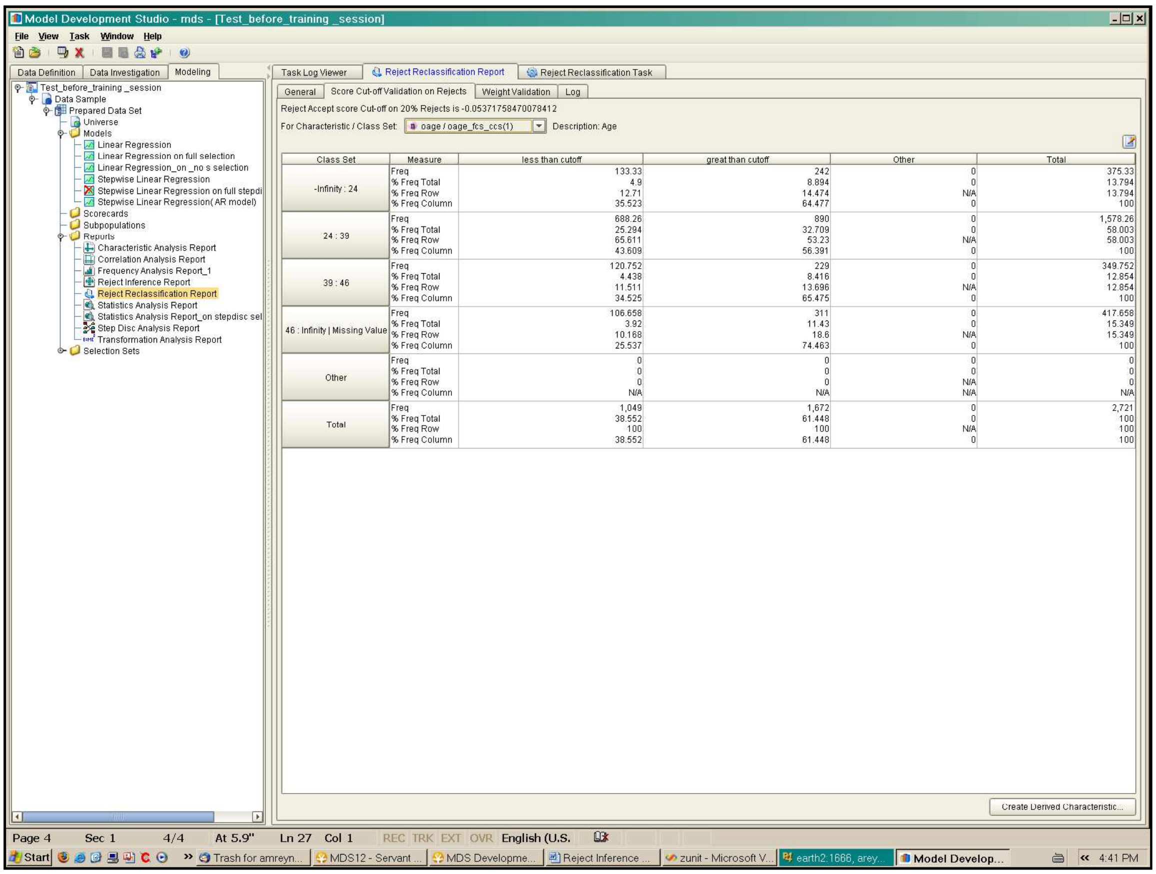
Assume 28% of Rejects have an RASCORE < Threshold, so the # of Reclassified Bad = 0.28*40=11.2. In order for the ratio of reclassified bad to the total known bad become 30/70 instead of 11.2/120, we need to weight reclassified bad by the value 30/70*120/11.2 = 4.59; so the final sample weights RAGTWGT for reclassified bad: 5*30/70*120/11.2 = 22.96.

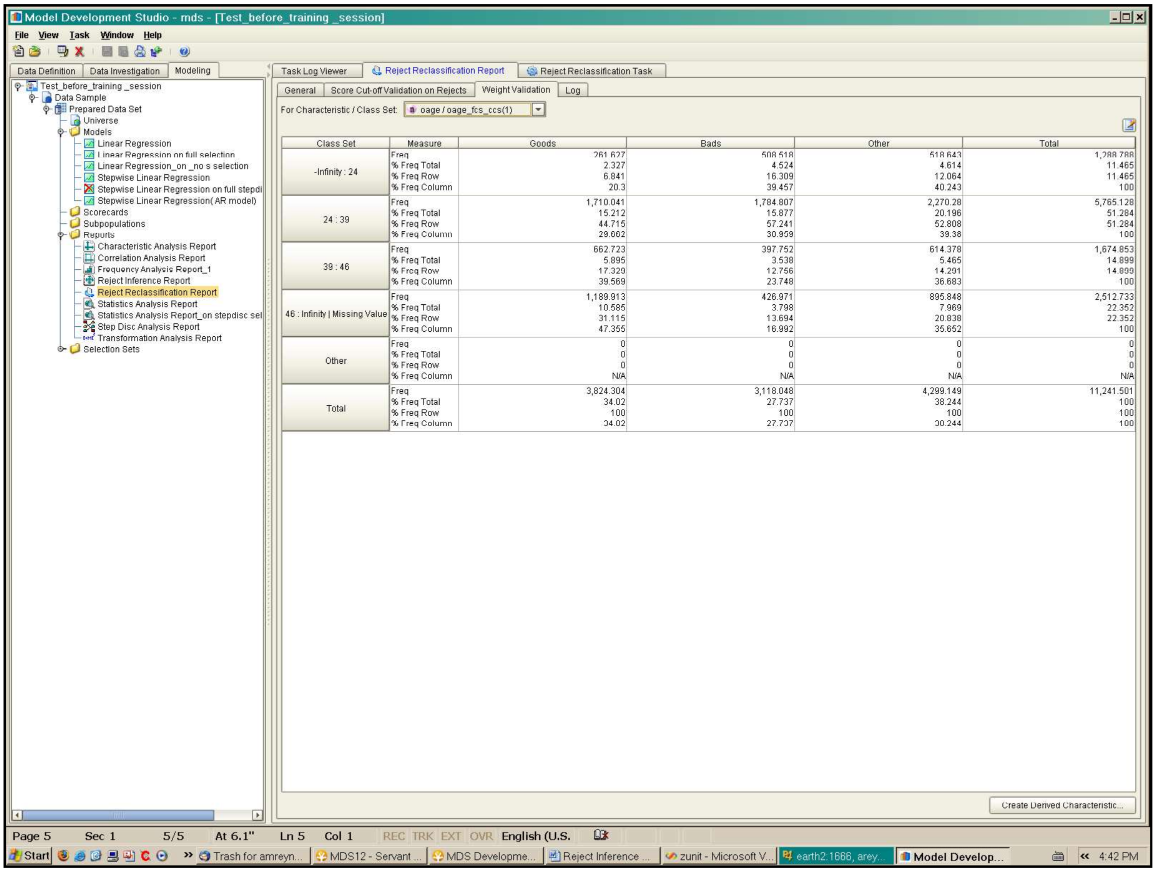
For new bad (BG1 = 0) weight BGEQWGT = RATWGT * (100 / 120) / (1 + 0.43) = RATWGT * 0.58

For every sample record, if RASCORE < Threshold, then BG1 = 0.

Upon performing reject inference, total number of bad (before taking sample weights into account) = 120+ 11.2*4.59 = 171.4; ratio of inferred bad to known (coming from accepts) bad = 11.2 * 4.59 / 120 = 0.43 = 30/70.

Appendix 1: Reject Reclassification Results




Model Development Studio - mds - [Test_before_training _session]
File View Task Window Help
Data Definition
Data Investigation
Modeling
Test_before_training _session
Data Sample
Prepared Data Set
Universe
Models
Linear Regression
Linear Regression on full selection
Linear Regression_on _no s selection
Stepwise Linear Regression
Stepwise Linear Regression on full stepdi
Stepwise Linear Regression(AR model)
Scorecards
Subpopulations
Reports
Characteristic Analysis Report
Correlation Analysis Report
Frequency Analysis Report_1
Reject Inference Report
Reject Reclassification Report
Statistics Analysis Report
Statistics Analysis Report_on stepdisc sel
Step Disc Analysis Report
Transformation Analysis Report
Selection Sets
Task Log Viewer
Reject Reclassification Report
Reject Reclassification Task
General
Score Cut-off Validation on Rejects
Weight Validation
Log
For Characteristic / Class Set: oage / oage_fcs_ccs(1)
Class Set | Measure | Goods | Bads | Other | Total
-Infinity : 24 | Freq | 261.627 | 508.518 | 518.643 | 1,288.788
% Freq Total | 2.327 | 4.524 | 4.614 | 11.465
% Freq Row | 6.841 | 16.309 | 12.064 | 11.465
% Freq Column | 20.3 | 39.457 | 40.243 | 100
24 : 39 | Freq | 1,710.041 | 1,784.807 | 2,270.28 | 5,765.128
% Freq Total | 15.212 | 15.877 | 20.196 | 51.284
% Freq Row | 44.715 | 57.241 | 52.808 | 51.284
% Freq Column | 29.662 | 30.959 | 39.38 | 100
39 : 46 | Freq | 662.723 | 397.752 | 614.378 | 1,674.853
% Freq Total | 5.895 | 3.538 | 5.465 | 14.899
% Freq Row | 17.329 | 12.756 | 14.291 | 14.899
% Freq Column | 39.569 | 23.748 | 36.683 | 100
46 : Infinity | Missing Value | Freq | 1,189.913 | 426.971 | 895.848 | 2,512.733
% Freq Total | 10.585 | 3.798 | 7.969 | 22.352
% Freq Row | 31.115 | 13.694 | 20.838 | 22.352
% Freq Column | 47.355 | 16.992 | 35.652 | 100
Other | Freq | 0 | 0 | 0 | 0
% Freq Total | 0 | 0 | 0 | 0
% Freq Row | 0 | 0 | 0 | 0
% Freq Column | N/A | N/A | N/A | N/A
Total | Freq | 3,824.304 | 3,118.048 | 4,299.149 | 11,241.501
% Freq Total | 34.02 | 27.737 | 38.244 | 100
% Freq Row | 100 | 100 | 100 | 100
% Freq Column | 34.02 | 27.737 | 30.244 | 100
Create Derived Characteristic...
Page 5 Sec 1 5/5 At 6.1" Ln 5 Col 1 REC TRK EXT OVR English (U.S.
Start
Trash for amreyn...
MDS12 - Servant ...
MDS Developme...
Reject Inference ...
zunit - Microsoft V...
earth2:1666, arey...
Model Develop...
4:42 PM