Building Better Credit Scores using Reject Inference and SAS

Steve Fleming, Clarity Services Inc.

ABSTRACT

Although acquisition credit scoring models are used to screen all applicants, the data available to create the scoring model typically only has outcomes for applicants who were previously approved for a loan (Siddiqi). Since approved applicants tend to be less risky than those that were previously rejected, building the acquisition score in this manner may produce biased results.

In this paper, four methods for dealing with missing outcome data are compared. The first, Ignore Rejects, uses only approved loans to build the model. The remaining three methods use a two-step approach where the model built on the approved loans is used to infer outcomes for the rejected applicants. A final model is then built using the known and inferred outcomes. The three methods evaluated here are Hard Cutoff, Parceling, and Individual. In this assessment, Parceling and Individual performed the best but, surprisingly, not much better than Ignore Rejects.

DATA

1,000 replications of 1,000 loan applications were created. Three intercorrelated predictor variables were created for each application.

pred1 ~ Normal(0,1)

pred2 ~ Normal(0,1) + 0.4*pred1

pred3 ~ Normal(0,1) + 0.4*pred2

```
pred1 = rand('NORMAL');
pred2 = rand('NORMAL') + 0.4*pred1 ;
pred3 = rand('NORMAL') + 0.4*pred2 ;
```

Then the probability of default and status were calculated.

logit = log odds of default = -0.6*pred1 - 0.4*pred2 - 0.2*pred3 – 2

pDefault = probability of default = exp(logit) / (1 + exp(logit))

default ~ Bernoulli(pDefault)

```
beta1 = -0.6; beta2 = -0.4;

beta3 = -0.2; /* predictor weights */

logit = beta1*pred1 + beta2*pred2 +
        beta3*pred3 - 2 ; /* log odds
        of default. */

prob_default = exp(logit) / (1 +
               exp(logit)); /*
               probability of default
               */

default = rand('BERNOULLI',
          prob_default);    /* randomly
          determined default status
          based on probability of
          default */
```

To simulate a decision system, applications were approved if any of the predictor variables exceeded a value of 2. This simulates a manual override of the decision system. Then, if any of the predictor variables were less than -1 the application was marked rejected. All remaining applications were marked approved.

```
if pred1 > 2 or pred2 > 2 or pred3 > 2
   then reject = 0; /* Override of
   decisioning */
else if pred1 < -1 or pred2 < -1 or
   pred3 < -1 then reject = 1; /*
   Normal reject decision */
else reject = 0; /* Normal approve
   decision */
```

Overall, 36.9% of applications were rejected. The default rate of approved applications was 9.16%. Normally, the default status of rejected applications would be unknown, but for this exercise, the default rate was 25.46%.

```
proc freq data=work.loan_performance ;
table reject*default / nopercent
    nocol;
run;
```

Simple statistics and correlation for the predictor variables are shown below.

```
proc corr data=work.loan_performance ;
var pred:;
run;
```

Simple Statistics

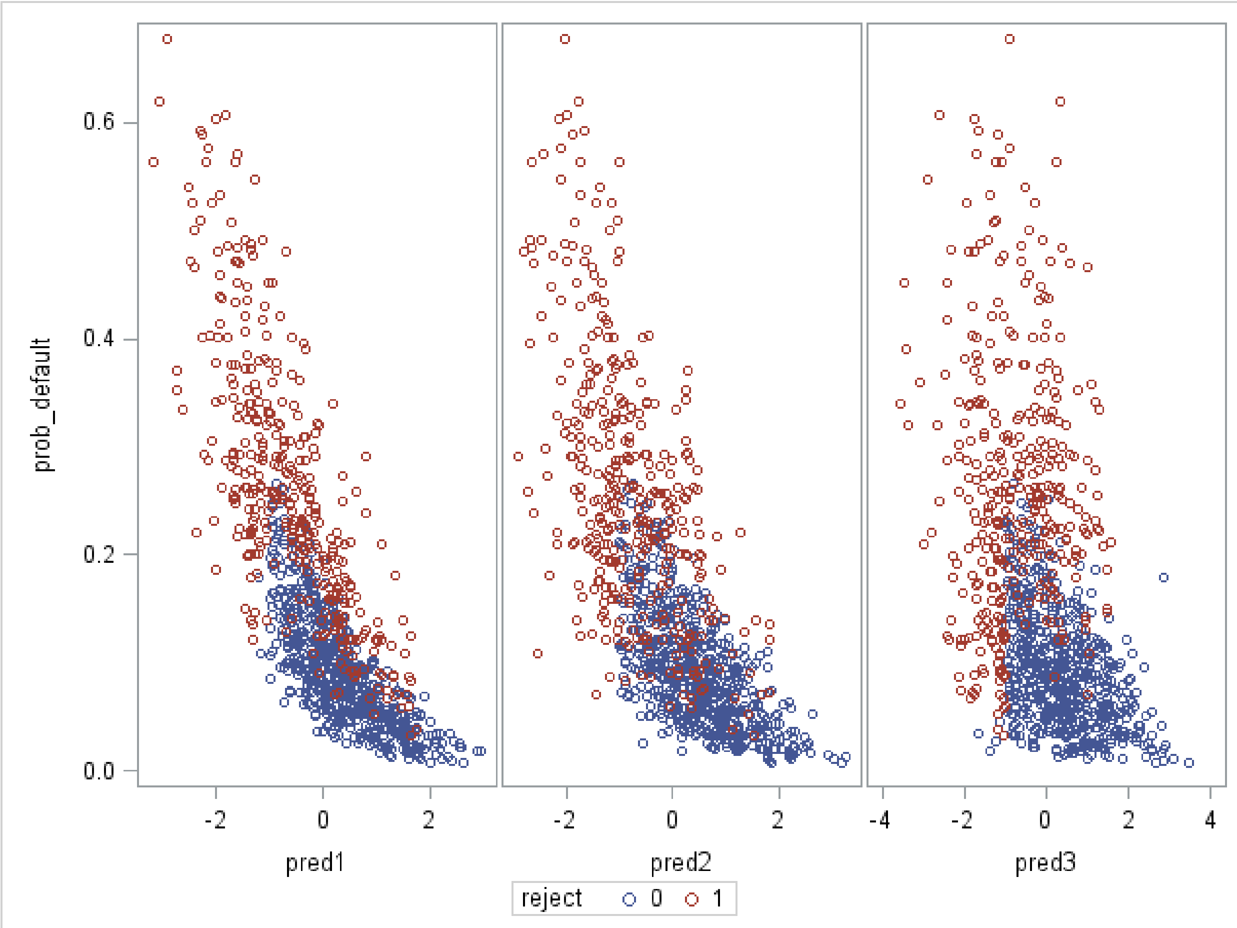
Variable	N	Mean	Std Dev	Minimum	Maximum
pred1	100,000	0.001	1.000	-4.88	4.88
pred2	100,000	0.003	1.076	-5.24	5.11
pred3	100,000	0.002	1.087	-5.12	5.27

Pearson Correlation Coefficients, N = 100000
Prob > |r| under H0: Rho=0

	pred1	pred2	pred3
pred1	1.000	0.370 <.0001	0.147 <.0001
pred2	0.370 <.0001	1.000	0.394 <.0001
pred3	0.147 <.0001	0.394 <.0001	1.000

In the following plot, the relationship between the predictors and the probability of default shows decreasing dependence for the predictors left to right. The chosen decision system results in few approved applications having a predictor value less than -1. A fair number of rejected applicants have a low probability of default. Few approved applicants have a probability of default greater than 0.2.

```
proc sgscatter
data=work.loan_performance ;
where rep=14;
compare x=(pred1 pred2 pred3)
y=prob_default / group=reject;
run;
```



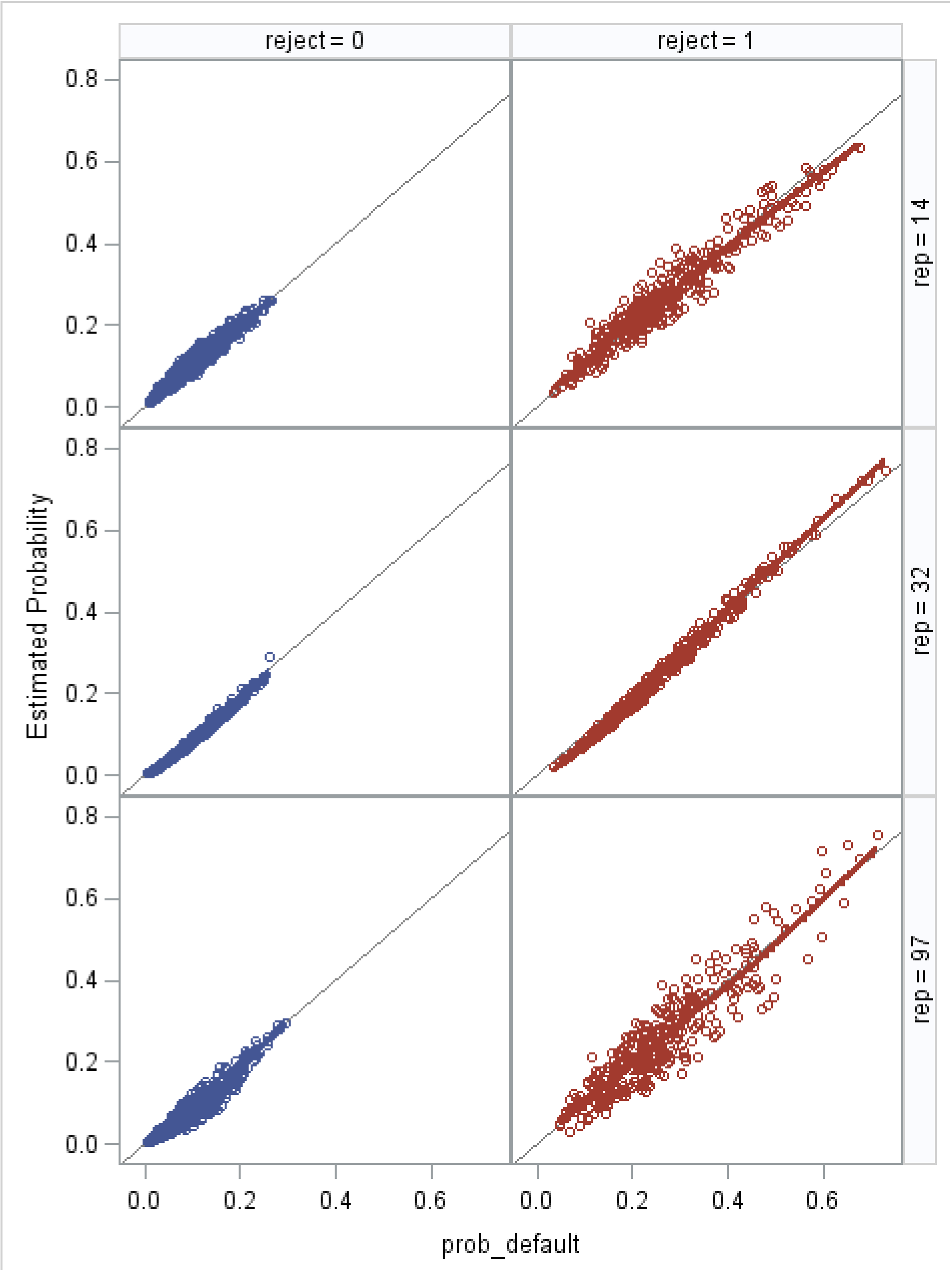
MODELING ALL DATA

A logistic regression model was fit to all of the data in each replication to give a baseline of what the results would look like if all loan applications were approved.

```
proc logistic data=work.loan_performance
  outest=work.all_est noprint;
by rep;
model default(event='1') = pred1 pred2 pred3 ;
output out=work.all_pred pred=p_1;
run;
```

In the following plot, the estimated probability of default closely matches the true probability of default.

```
proc sgpanel data=work.all_pred noautolegend;
where rep in (14,32,97);
panelby reject rep  / layout=lattice;
lineparm x=0 y=0 slope=1 / lineattrs=(color=grey);
scatter x=prob_default y=p_1 / group=reject ;
loess x=prob_default y=p_1 / group=reject
  lineattrs=(thickness=3);
run;
```



Most importantly for credit scoring, the rank-order correlation between the true and estimated probability of default across the replications is very close to 1 in most cases.

```
proc corr data=work.all_pred spearman noprint
   outs=work.all_pred_corrs
   (where=(_name_='prob_default'));
by rep;
var prob_default p_1;
run;

proc means data=work.all_pred_corrs min p25 p50 p75
   max maxdec=3;
var p_1;
run;
```

Minimum	25th Pctl	50th Pctl	75th Pctl	Maximum
0.906	0.985	0.992	0.997	1.000

IGNORE REJECTS

A logistic regression model was fit to all approved applications.

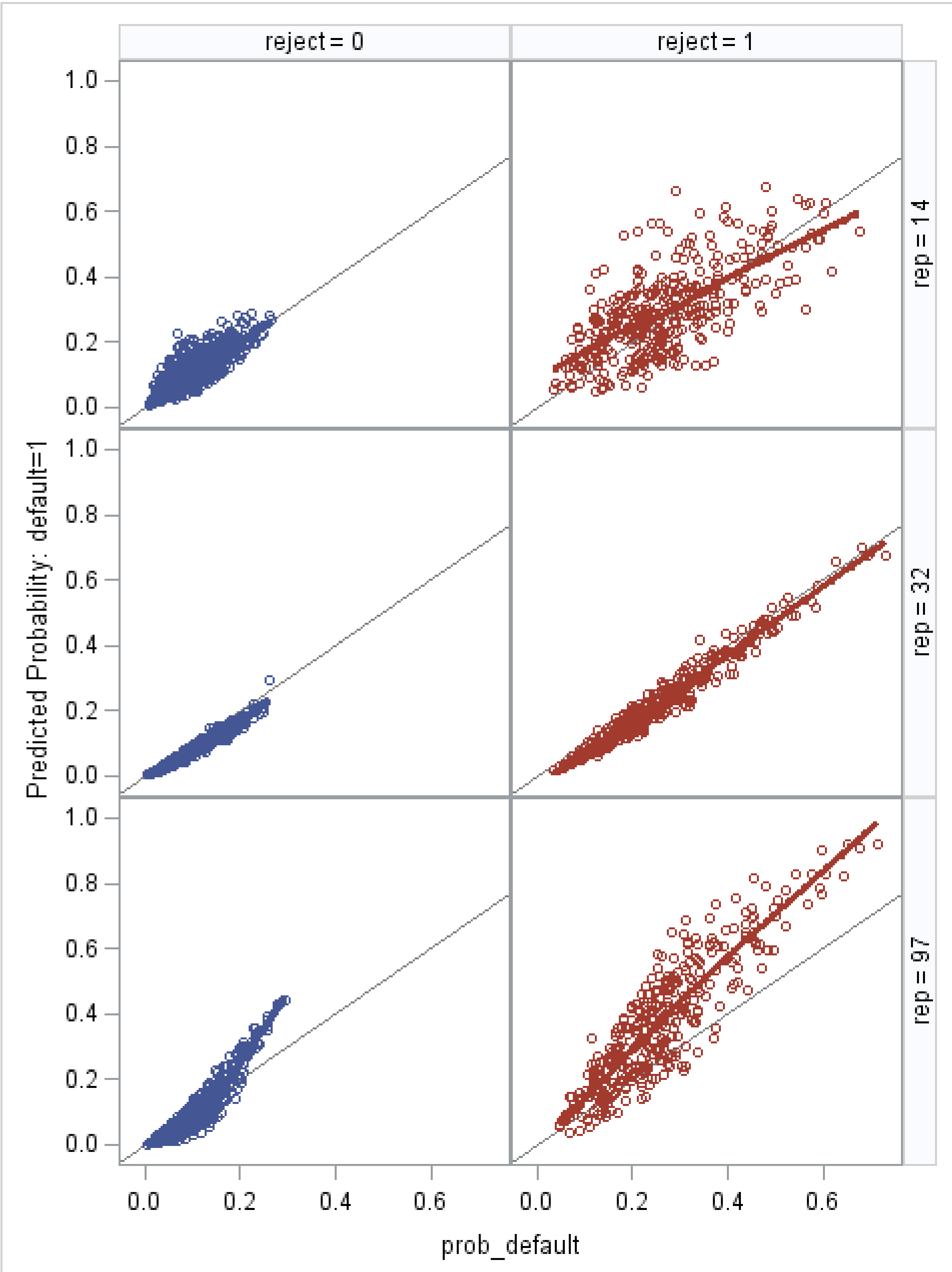
```
proc logistic data=work.loan_performance
   outest=work.acc_est outmodel=work.acc_model
   noprint ;
by rep;
where reject=0;
model default(event='1') = pred1 pred2 pred3 ;
output out=work.acc_pred pred=p_1;
run;
```

This model was then used to estimate the probability of default for the rejected applications. It is expected that this inference will be biased due to prediction outside the range of the data used to estimate the model.

```
proc logistic inmodel=work.acc_model;
by rep;
score data=work.loan_performance
   (where=(reject=1))
  out=work.rej_scored_w_acc_model;
run;
```

After putting the approved and rejected application back together, the following plot demonstrates that the estimated probability of default does not match the true probability of default closely for rejects when rejects are ignored in the model development. Some replications seem to fit better than others.

```
data work.ignore_rejects;
set work.rej_scored_w_acc_model
work.acc_pred(where=(reject=0));
run;
```



The rank-order correlations when rejects are ignored are not as close to 1 across the replications even dropping below 0.9 for some.

Minimum	25th Pctl	50th Pctl	75th Pctl	Maximum
0.683	0.949	0.974	0.990	1.000

HARD CUTOFF

Approaches to use inference to allow rejected applications to influence the model are called reject inference. The simplest

```
/* Calculate the default rate in each replication */
proc summary
    data=work.loan_performance(where=(reject=0)) nway
    ;
```

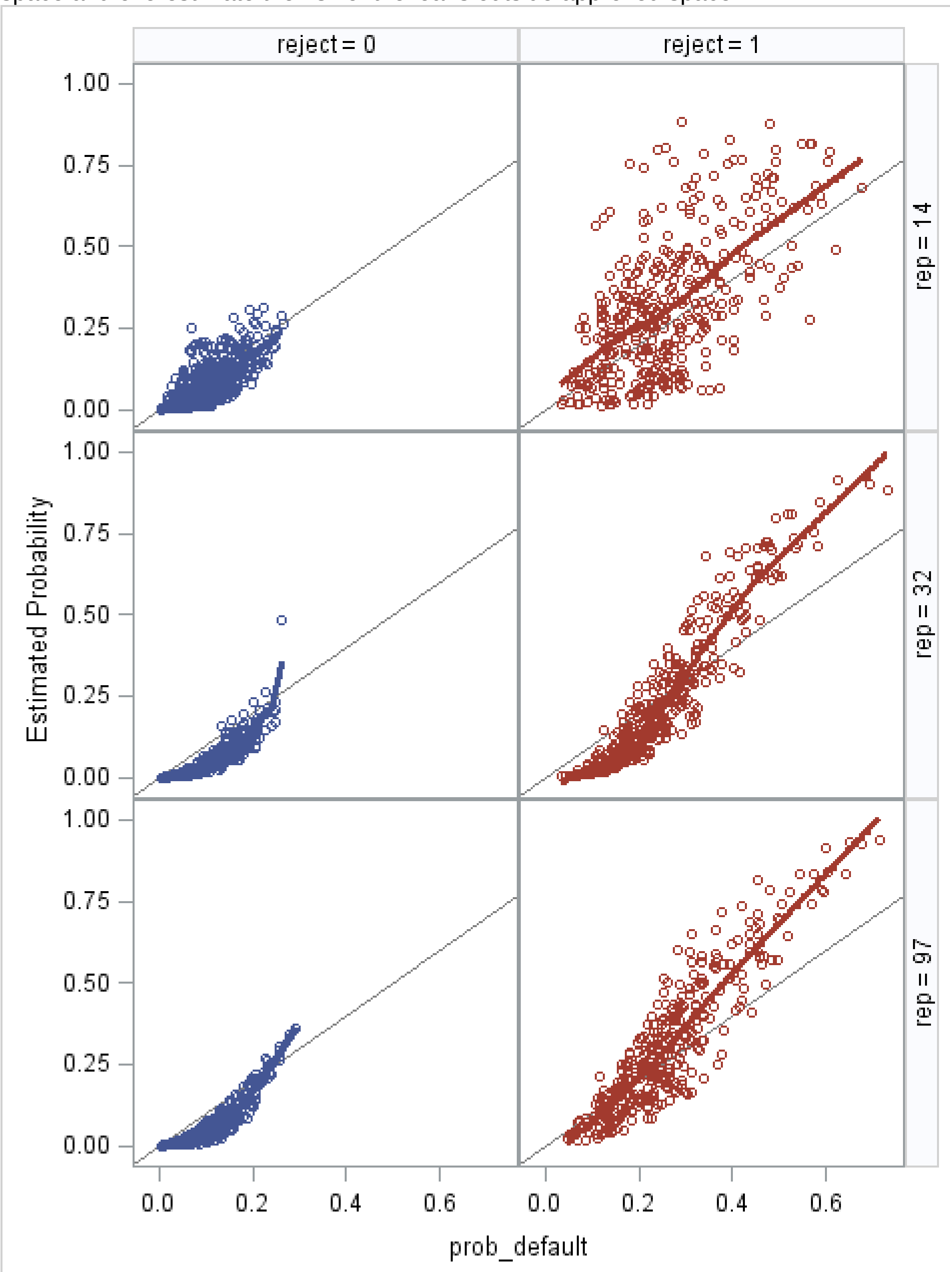
reject inference is Hard Cutoff. Using the logistic regression model fit to approved applications, the rejected applications are scored. It is assumed that the rejected applications will have 2 to 4 times the default rate of approved loans (Siddiqi). We use 3 times for this exercise.

```
by rep;
var default;
output out=work.acc_default_rates mean=default_rate ;
run;

/* Triple the default odds for rejects */
data work.hard_cutoff (keep=rep adjusted_prob
    expected_defaults);
set work.acc_default_rates;
adjusted_odds = (default_rate / (1 - default_rate)) *
    3;
adjusted_prob = adjusted_odds / (adjusted_odds + 1);
expected_defaults = (adjusted_prob) * (&nApps -
    _freq_);
run;
```

The scored rejects are then sorted and the ones with the highest estimated probability of default are inferred to be defaults until enough defaults have been assigned to make the default rate for the rejects bad enough.

```
proc sort data=work.rej_scored_w_acc_model;
by rep descending p_1;
run;

/* Mark the rejects with the highest estimated
    probability of default as defaults until the
    expected
     number of defaults is reached */
data work.rej_hc_result;
merge work.rej_scored_w_acc_model work.hard_cutoff ;
by rep;
retain rep_cnt .;
if first.rep then rep_cnt = 0;
rep_cnt + 1;
if rep_cnt < expected_defaults then default_hc = 1;
else default_hc = 0;
run;

/* Combine rejects with inferred outcomes with
   approved loans */
data work.reject_inference_hc;
set work.rej_hc_result (drop=p_1 in=rej)
   work.loan_performance (in=acc where=(reject=0));
if acc then default_hc = default;
run;
```

As shown in the following plot, using Hard Cutoff appears to underestimate the risk of the loans inside the approved space and overestimate the risk of the loans outside approved space.



The rank-order correlation between the true and estimated probability of default across the replications appears to be worse for Hard Cutoff than simply ignoring rejects.

Minimum	25th Pctl	50th Pctl	75th Pctl	Maximum
0.603	0.919	0.957	0.982	1.000

PARCELING

In parceling reject inference the rejects are split into risk bands

```
/* Break approved applications within each
   replication into quintile risk bands */
```

based on the initial model.

```
proc univariate data=work.acc_pred noprint;
by rep;
var p_1 ;
output out=work.acc_deciles pctlpre=P_ pctlpts= 20
  to 80 by 20;
run;

proc transpose data=work.acc_deciles
  out=work.acc_deciles_t;
by rep;
run;

/* Create formats tied to the quintile risk bands */
data work.acc_cntlin (keep=start end label fmtname
  type);
set work.acc_deciles_t end=last ;
by rep;
length startx endx $4 label $9 fmtname $6;
retain end . endx 'zzzz' fmtname ' ' type 'n';

if first.rep then do;
    start = 0;
    startx = 'Min';
    fmtname = cats('a', put(rep,z4.), 'd');
end;
else do;
    start = end ;
    startx = endx;
end;

end = col1 ;
endx = strip(_name_) ;
label = cats(startx, '-', endx);
output;

if last.rep then do;
    start = end;
    startx = endx;
    end = 1;
    endx = 'Max';
    label = cats(startx, '-', endx);
    output;
end;
run;

proc format cntlin=work.acc_cntlin;
run;

data work.acc_parcel;
set work.acc_pred;
length parcel_group $9 fmt $7 ;
fmt = cats('a', put(rep,z4.), 'd.');
parcel_group = strip(putn(p_1, fmt));
run;

/* Calculate observed default rates within quintile
  risk bands */
```

```
proc summary data=work.acc_parcel nway;
class rep parcel_group;
var default;
output out=work.acc_decile_default mean=p_default ;
run;
```

As with Hard Cutoff, It is assumed that the rejected applications will have a higher default rate than approved applications. The adjustment this time is made within risk bands.

```
/* Triple the default odds for rejects */
data work.parceling (keep=rep parcel_group
  adjusted_prob);
set work.acc_decile_default;
adjusted_odds = (p_default / (1 - p_default)) * 3;
adjusted_prob = adjusted_odds / (adjusted_odds + 1);
run;

data work.rej_parcel (keep=rep parcel_group pred:
  logit prob_default default reject );
set work.rej_scored_w_acc_model;
length parcel_group $9 fmt $7 ;
fmt = cats('a', put(rep,z4.), 'd.');
parcel_group = strip(putn(p_1, fmt));
run;

proc freq data=work.rej_parcel noprint;
by rep;
table parcel_group / out=work.parcel_counts
  (drop=percent);
run;

data work.rej_parcel_exp_defaults (keep=rep
  parcel_group expected_defaults);
merge work.parceling work.parcel_counts (in=pc) ;
by rep parcel_group ;
if pc;
expected_defaults = count * adjusted_prob ;
run;

proc sort data=work.rej_parcel;
by rep parcel_group;
run;
```

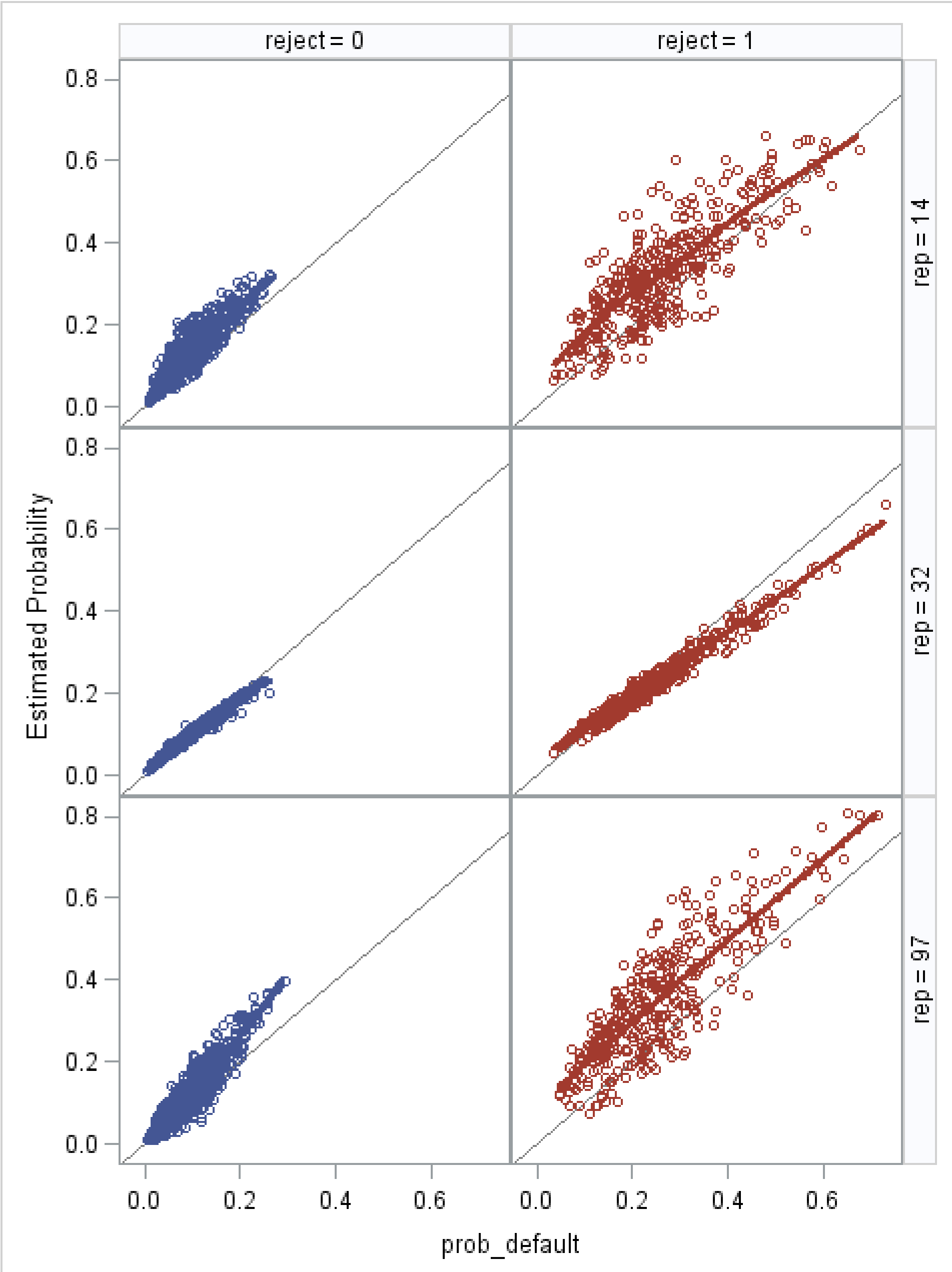
Randomly selected rejects within risk bands are inferred to be defaults until enough defaults have been assigned to make the default rate for the rejects bad enough within each risk band.

```
data work.rej_parcel_w_exp_def;
merge work.rej_parcel work.rej_parcel_exp_defaults ;
by rep parcel_group;
CALL STREAMINIT(343473193);
sortkey = rand('UNIFORM');
run;

proc sort data=work.rej_parcel_w_exp_def;
by rep parcel_group sortkey;
run;

data work.rej_parc_result;
set work.rej_parcel_w_exp_def;
by rep parcel_group;
retain group_cnt .;
if first.parcel_group then group_cnt = 0;
group_cnt + 1;
```

```
if group_cnt < expected_defaults then default_parc =
  1;
else default_parc = 0;
run;
```

The plot on the next page shows that using Parceling appears to provide a better estimate of the risk inherent in rejected applications than Hard Cutoff did. In addition, the rank-order correlations are much closer to 1 than when using Hard Cutoff.

Minimum	25th Pctl	50th Pctl	75th Pctl	Maximum
0.811	0.959	0.981	0.992	1.000



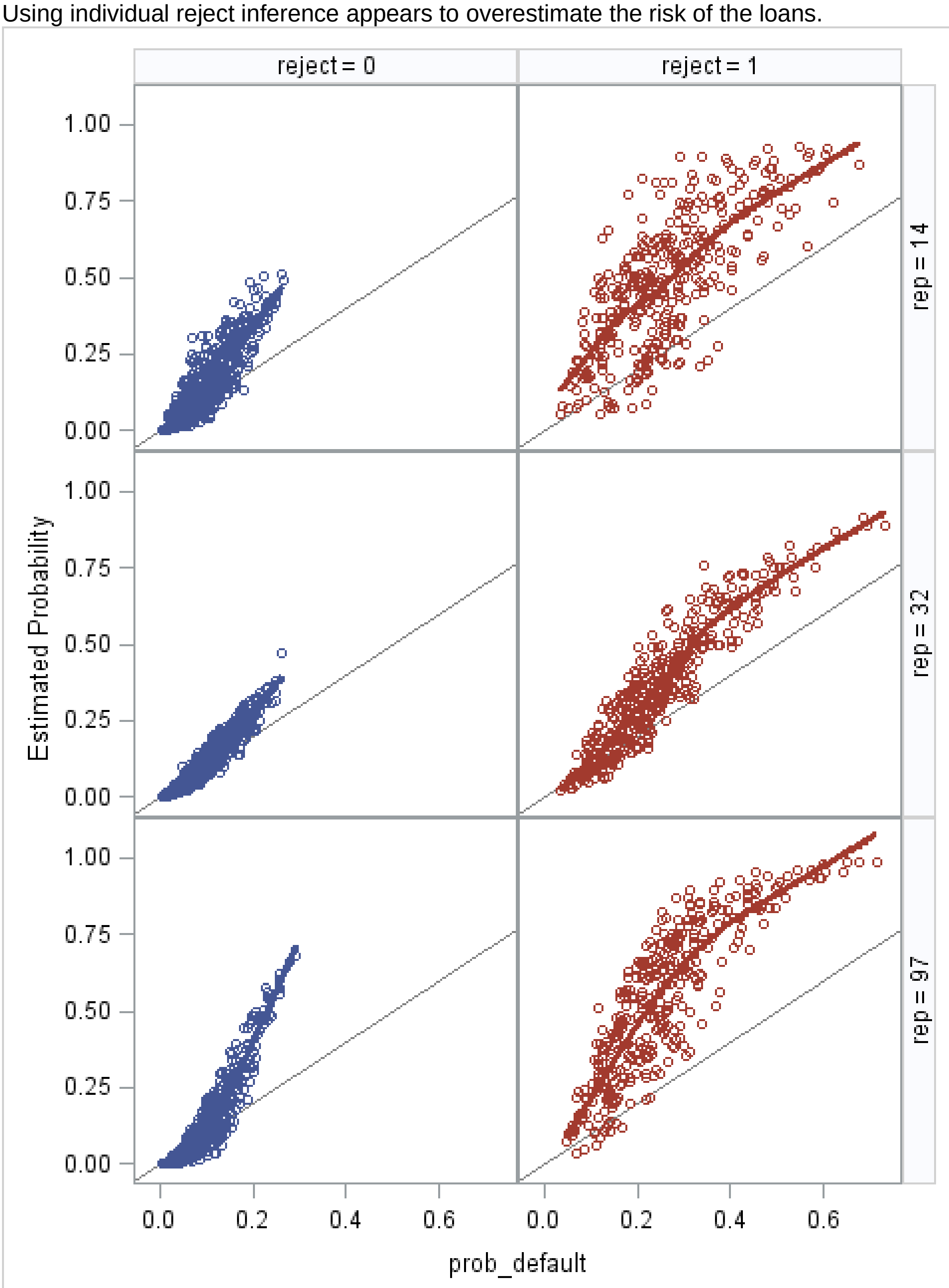
INDIVIDUAL

Logistic regression models the probability of an occurrence. In Individual reject inference the estimated probability of defaults from the logistic regression model built on approved applications are adjusted to make the rejects riskier. Each reject is independently inferred a default status based on the adjusted probability.

```
data work.individual (drop=p_1);
set work.rej_scored_w_acc_model (in=rej)
     work.loan_performance (in=acc where=(reject=0))
     ;
if acc then default_ind = default;
else if rej then do;
    /* Triple the default odds for rejects */
    adjusted_odds = (p_1 / (1 - p_1)) * 3;
    adjusted_prob = adjusted_odds / (adjusted_odds +
     1);
```

```
        /* infer performance */
        default_ind = rand('Bernoulli', adjusted_prob);
    end;
    run;
```

Using individual reject inference appears to overestimate the risk of the loans.



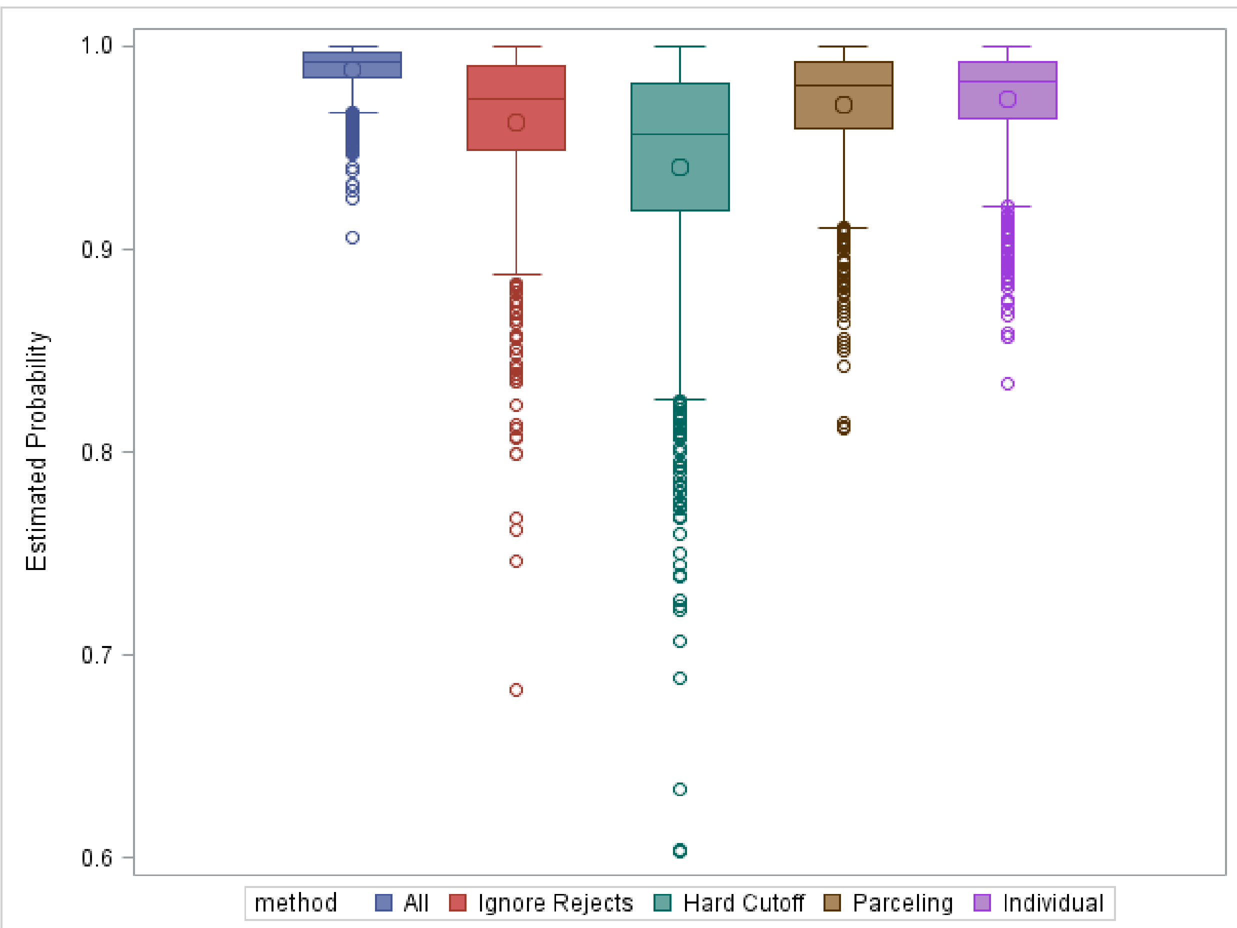
However, the rank-order correlations appear to be on par with Parceling.

Minimum	25th Pctl	50th Pctl	75th Pctl	Maximum
0.834	0.964	0.982	0.992	1.000

COMPARISON OF REJECT INFERENCE METHODS

For this comparison, Hard Cutoff reject inference performed even worse than Ignoring Rejects. Individual reject inference gave the most consistent rank-order correlations although Parceling was not far behind. Either of those methods are preferable to ignoring reject altogether.

```
data work.comparison (keep=method p_1);
set work.all_pred_corrs (in=al)
    work.ignore_rejects_corrs (in=no)
    work.reject_inference_hc_corrs (in=hc)
    work.reject_inference_parc_corrs (in=pa)
    work.reject_inference_ind_corrs (in=in)
    ;
length method $17;
if al then method = 'All';
else if no then method = 'Ignore Rejects';
else if hc then method = 'Hard Cutoff';
else if pa then method = 'Parceling';
else if in then method = 'Individual';
run;

proc sgplot;
vbox p_1 / group=method;
run;
```



CONCLUSION

It is clear from this study that Hard Cutoff reject inference suffers the most issues of the attempted methods. To preserve the rank order of the true probabilities of default, either Parceling or Individual reject inference may be suitable.

REFERENCES

Siddiqi, Naeem. 2006. *Credit Risk Scorecards*. Hoboken, New Jersey: John Wiley & Sons

ACKNOWLEDGMENTS

I would like to thank the Analytics Team at Clarity Services for their help improving this work.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at sfleming@clarityservices.com.

Complete code for reproducing these results is available at:

https://drive.google.com/file/d/0B-QzBZnCSTw4SGFiZVhkU2tNVWM/view?usp=sharing