```
/*************************************************************
*************************************************************
*************************************************************
*************************************************************/
/*Daren Purnell, 2017SP_PREDICT_411-DL_SEC60*/
/* Connect Predict411 Data */
libname mydata "/scs/wtm926/" access=readonly;
proc datasets library=mydata;
run;
ods graphics on;
title 'Wine Sales Project';
/* Create a shortcut to wine sales data */
data base_data;
   set mydata.wine;
run;
proc contents data = base_data; run;
proc print data = base_data (obs=10); run; quit;
/*EXPLORATORY DATA ANALYSIS*/
/* Copy over base data set for EDA */
data eda_data;
        set base_data;
        TARGET_FLAG = ( TARGET > 0 ); /* 1 if cases sold; 0 if no cases sold */
        TARGET_AMT = TARGET - 1;
        if TARGET_FLAG = 0 then TARGET_AMT = .;
run;
proc print data = eda_data (obs=10); run; quit;
/* VAR = 1.55 is 1/2 of Mean = 3.85 when TARGET > 0, Pretty close.
I think either NB or POI will work */
proc univariate normal plot data = eda_data;
        var TARGET;
        histogram TARGET /normal midpoints = 0 1 2 3 4 5 6 7 8 9;
run;
proc univariate data = eda_data;
        var TARGET_AMT;
        histogram TARGET_AMT /normal midpoints = 0 1 2 3 4 5 6 7 8 9;
run;
proc sort data = eda_data; by TARGET_FLAG; run;
proc freq data = eda_data;
        tables TARGET_FLAG/ missing;
run;
proc means data = eda_data n nmiss mean mode var Q1 Q3 max min;
        where TARGET > 0;
        var TARGET FixedAcidity        VolatileAcidity CitricAcid ResidualSugar Chlorides
FreeSulfurDioxide
```

```
        TotalSulfurDioxide Density pH Sulphates      Alcohol          LabelAppeal    AcidIndex
STARS;
run;
proc univariate normal plot data = eda_data;
        title 'Fixed Acidity: Fixed Acidity of Wine';
        var FixedAcidity;
        histogram FixedAcidity/normal;
run;
proc univariate normal plot data = eda_data;
        title 'VolatileAcidity: Volatile Acid Content of Wine';
        var VolatileAcidity;
        histogram VolatileAcidity/normal;
run;
proc univariate normal plot data = eda_data;
        title 'CitricAcid: Citric Acid Content of Wine';
        var CitricAcid;
        histogram CitricAcid/normal;
run;
proc univariate normal plot data = eda_data;
        title 'ResidualSugar: Residual Sugar of Wine';
        var ResidualSugar;
        histogram ResidualSugar/normal;
run;
proc univariate normal plot data = eda_data;
        title 'Chlorides: Chloride Content of Wine';
        var Chlorides;
        histogram Chlorides/normal;
run;
proc univariate normal plot data = eda_data;
        title 'FreeSulfurDioxide: Sulfur Dioxide Content of Wine';
        var FreeSulfurDioxide;
        histogram FreeSulfurDioxide/normal;
run;
proc univariate normal plot data = eda_data;
        title 'TotalSulfurDioxide: Sulfur Dioxide Content of Wine';
        var TotalSulfurDioxide;
        histogram TotalSulfurDioxide/normal;
run;
proc univariate normal plot data = eda_data;
        title 'Density: Density of Wine (thickness)';
        var Density;
        histogram Density/normal;
run;
proc univariate normal plot data = eda_data;
```

```
        title 'pH: pH of Wine (Acid or Base)';
        var pH;
        histogram pH/normal;
run;
proc univariate normal plot data = eda_data;
        title 'Sulphates: Sulfate Content of Wine';
        var Sulphates;
        histogram Sulphates/normal;
run;
proc univariate normal plot data = eda_data;
        title 'Alcohol: Alcohol Content of Wine';
        var Alcohol;
        histogram Alcohol/normal;
run;
proc univariate normal plot data = eda_data;
        title 'LabelAppeal: High (+) Customers like the label; (-) Dislike of Label';
        var LabelAppeal;
        histogram LabelAppeal/normal;
run;
proc univariate normal plot data = eda_data;
        title 'AcidIndex: Method of testing total Acidity of Wine by using a weighted avg.';
        var AcidIndex;
        histogram AcidIndex/normal;
run;
proc univariate normal plot data = eda_data;
        title 'STARS: Expert Wine Rating. 4=Excellent 1= Poor';
        var STARS;
        histogram STARS/normal;
run;
/* STARS and LABEL APPEAL can be treated as CATEGORICAL/ORDINAL */
proc sort data = eda_data; by STARS; run;
proc freq data = eda_data;
        tables STARS*TARGET/ missing;
run;
proc sort data = eda_data; by LABELAPPEAL; run;
proc freq data = eda_data;
        tables LABELAPPEAL*TARGET/ missing;
run;
proc sort data = eda_data; by ACIDINDEX; run;
proc freq data = eda_data;
        tables ACIDINDEX*TARGET/ missing;
run;
/*Investigate relationships between target variables and regressors*/
proc corr data = eda_data;
```

```
        with FixedAcidity VolatileAcidity CitricAcid ResidualSugar Chlorides FreeSulfurDioxide
        TotalSulfurDioxide Density pH Sulphates      Alcohol          LabelAppeal    AcidIndex
STARS;
        var TARGET TARGET_FLAG TARGET_AMT;
run;
/*IMPUTATIONS*/
data imp_data;
        set eda_data;
        IMP_STARS                       = STARS;
        IMP_Sulphates                   = Sulphates;
        IMP_Alcohol                     = Alcohol;
        IMP_TotalSulfurDioxide      = TotalSulfurDioxide;
        IMP_Chlorides               = Chlorides;
        IMP_FreeSulfurDioxide   = FreeSulfurDioxide;
        IMP_ResidualSugar           = ResidualSugar;
        IMP_pH                              = pH;

        /*Missing values for STARS seems significant, based off PROC FREQ, and warrants a
flag*/
        F_STARS                             = 0;

        if missing(STARS)                   then do;IMP_STARS           = 1;
F_STARS = 1; end;
        if missing(Sulphates)           then IMP_Sulphates              = 0.5271118;
        if missing(Alcohol)                 then IMP_Alcohol                    = 9.4;
        if missing(TotalSulfurDioxide)then IMP_TotalSulfurDioxide = 125.0000000;
        if missing(Chlorides)           then IMP_Chlorides              = 0.0548225;
        if missing(FreeSulfurDioxide)  then IMP_FreeSulfurDioxide  = 30.8455713;
        if missing(ResidualSugar)       then IMP_ResidualSugar          = 5.4187331;
        if missing(pH)                      then IMP_pH
        = 3.2076282;

        keep TARGET Index
                TARGET_FLAG
                TARGET_AMT
                AcidIndex
                IMP_Alcohol
                IMP_Chlorides
                CitricAcid
                Density
                FixedAcidity
                IMP_FreeSulfurDioxide
                LabelAppeal
                IMP_ResidualSugar
```

```
                IMP_STARS
                F_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                VolatileAcidity
                IMP_pH
        ;
run;
proc print data = imp_data (obs=10); run; quit;
/*TRANSFORMATIONS*/
/* 1. Absolute values of all negative concentration values. 2. Xfer distro to adhere to
normaility*/
data xfer_data;
        set imp_data;
        FixedAcidity = sqrt(abs(FixedAcidity) + 1);
        VolatileAcidity = log(abs(VolatileAcidity));
        CitricAcid = sqrt(abs(CitricAcid));
        IMP_ResidualSugar = log(abs(IMP_ResidualSugar)+ 1);
        IMP_Chlorides = sqrt(abs(IMP_Chlorides));
        IMP_FreeSulfurDioxide = log(abs(IMP_FreeSulfurDioxide) + 1) ;
        IMP_TotalSulfurDioxide = log(abs(IMP_TotalSulfurDioxide) + 1);
run;
/* Verify effects of transformations */
proc univariate data = xfer_data;
        title ' FixedAcidity Xfer ';
        var FixedAcidity;
        histogram FixedAcidity/normal;
run;
proc univariate data = xfer_data;
        title 'VolatileAcidity Xfer ';
        var VolatileAcidity;
        histogram VolatileAcidity/normal;
run;
proc univariate data = xfer_data;
        title 'CitricAcid Xfer ';
        var CitricAcid;
        histogram CitricAcid/normal;
run;
proc univariate data = xfer_data;
        title 'IMP_ResidualSugar Xfer ';
        var IMP_ResidualSugar;
        histogram IMP_ResidualSugar/normal;
run;
proc univariate data = xfer_data;
```

```
        title 'IMP_ResidualSugar Xfer ';
        var IMP_ResidualSugar;
        histogram IMP_ResidualSugar/normal;
run;
proc univariate data = xfer_data;
        title ' IMP_Chlorides Xfer ';
        var IMP_Chlorides;
        histogram IMP_Chlorides/normal;
run;
proc univariate data = xfer_data;
        title 'IMP_FreeSulfurDioxide Xfer ';
        var IMP_FreeSulfurDioxide;
        histogram IMP_FreeSulfurDioxide/normal;
run;
proc univariate data = xfer_data;
        title 'IMP__TotalSulfurDioxide Xfer ';
        var IMP_TotalSulfurDioxide;
        histogram IMP_TotalSulfurDioxide/normal;
run;
/*MODEL BUILDING*/
/* TARGET Model Adj R^2: 0.5390 VAR: AcidIndex IMP_Alcohol IMP_Chlorides CitricAcid
Density
IMP_FreeSulfurDioxide LabelAppeal IMP_STARS F_STARS IMP_Sulphates
IMP_TotalSulfurDioxide VolatileAcidity IMP_pH */
proc reg data = xfer_data;
        model TARGET = AcidIndex
                IMP_Alcohol
                IMP_Chlorides
                CitricAcid
                Density
                FixedAcidity
                IMP_FreeSulfurDioxide
                LabelAppeal
                IMP_ResidualSugar
                IMP_STARS
                F_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                VolatileAcidity
                IMP_pH/ vif selection = stepwise;
run;
/* Utilize auto variable selection in OLS regression to select variables for predicting
TARGET/TARGET_AMT. Adj R^2:
```

Stepwise:0.5984 Backwards:0.5984 Forward: 0.5984; All the same. Stepwise selected for TARGET_AMT:
AcidIndex IMP_Alcohol IMP_Chlorides Density IMP_FreeSulfurDioxide LabelAppeal IMP_STARS F_STARS
VolatileAcidity IMP_pH*/

```
proc reg data = xfer_data;
        model TARGET_AMT = AcidIndex
                IMP_Alcohol
                IMP_Chlorides
                CitricAcid
                Density
                FixedAcidity
                IMP_FreeSulfurDioxide
                LabelAppeal
                IMP_ResidualSugar
                IMP_STARS
                F_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                VolatileAcidity
                IMP_pH/ vif selection = stepwise;
run;
```

/*Utilize auto variable selection in OLS regression to select variables for predicting TARGET_FLAG
Stepwise:0.3979 Backwards:0.3979 Forward:0.3979; all the same. Stepwise Selected: AcidIndex IMP_Alcohol CitricAcid IMP_FreeSulfurDioxide LabelAppeal IMP_ResidualSugar IMP_STARS F_STARS
IMP_Sulphates IMP_TotalSulfurDioxide VolatileAcidity IMP_pH */

```
proc reg data = xfer_data;
        model TARGET_FLAG = AcidIndex
                IMP_Alcohol
                IMP_Chlorides
                CitricAcid
                Density
                FixedAcidity
                IMP_FreeSulfurDioxide
                LabelAppeal
                IMP_ResidualSugar
                IMP_STARS
                F_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                VolatileAcidity
                IMP_pH/ vif selection = stepwise;
```

```
run;
data model_data;
	set xfer_data;
run;
proc print data = model_data (obs=10); run;
/* Poisson: Using Variables REG VAR Selection AIC 1:51394. (w/CLASS) , 2:45647(w/o CLASS) */
proc genmod data = model_data;
	model TARGET = AcidIndex
				IMP_Alcohol
				IMP_Chlorides
				CitricAcid
				Density
				IMP_FreeSulfurDioxide
				IMP_STARS
				F_STARS
				LabelAppeal
				IMP_Sulphates
				IMP_TotalSulfurDioxide
				VolatileAcidity
				IMP_pH
		/ dist=poi link=log;
		output out=model_data pred=P_TARGET_POI;
run;
/* Hurdle POI TARGET_AMT */
proc genmod data = model_data;
	model TARGET_AMT = AcidIndex
					IMP_Alcohol
					CitricAcid
					IMP_FreeSulfurDioxide
					LabelAppeal
					IMP_ResidualSugar
					IMP_STARS
					F_STARS
					IMP_Sulphates
					IMP_TotalSulfurDioxide
					VolatileAcidity
					IMP_pH
		/ dist=poi link=log;
		output out=model_data pred=P_TARGET_AMT_POI;
run;
/* NB */
/* Review Results: Both models produce the exact same results with same variables.
  Adding all variables to NB for comparison AIC 1:51397 (w/CLASS) 2:45652 (w/o CLASS)*/
proc genmod data = model_data;
```

```
        model TARGET = AcidIndex
                IMP_Alcohol
                IMP_Chlorides
                CitricAcid
                Density
                FixedAcidity
                IMP_STARS
                F_STARS
                IMP_FreeSulfurDioxide
                LabelAppeal
                IMP_ResidualSugar
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                VolatileAcidity
                IMP_pH
               / dist=NB link=log;
               output out=model_data pred=P_TARGET_NB;
run;
proc print data = model_data (obs=10);
        var TARGET P_TARGET_POI P_TARGET_NB;
run;
/* Logistic Regression to model TARGET_FLAG AUC = .8989 */
proc logistic data = model_data plot(only)=(roc(ID=prob));
        model TARGET_FLAG(ref='0') = AcidIndex
                        IMP_Alcohol
                        CitricAcid
                        IMP_FreeSulfurDioxide
                        LabelAppeal
                        IMP_STARS
                        F_STARS
                        IMP_ResidualSugar
                        IMP_Sulphates
                        IMP_TotalSulfurDioxide
                        VolatileAcidity
                        IMP_pH/roceps=0.1;
        output out=model_data pred=P_ZERO_LOG;
run;
/* ZIP Model */
proc genmod data = model_data;
        model TARGET = AcidIndex
                                IMP_Alcohol
                                IMP_Chlorides
                                CitricAcid
                                Density
```

```
                                IMP_FreeSulfurDioxide
                                LabelAppeal
                                IMP_STARS
                                F_STARS
                                IMP_Sulphates
                                IMP_TotalSulfurDioxide
                                VolatileAcidity
                                IMP_pH
                        / dist=ZIP link=log;
        zeromodel AcidIndex
                                IMP_Alcohol
                                CitricAcid
                                IMP_FreeSulfurDioxide
                                LabelAppeal
                                IMP_ResidualSugar
                                IMP_STARS F_STARS
                                IMP_Sulphates
                                IMP_TotalSulfurDioxide
                                VolatileAcidity
                                / link=logit;
                                output out=model_data pred=P_TARGET_ZIP pzero=P_ZERO_ZIP;
run;
/* ZINB Model */
proc genmod data = model_data;
        model TARGET = AcidIndex
                                IMP_Alcohol
                                IMP_Chlorides
                                CitricAcid
                                Density
                                IMP_FreeSulfurDioxide
                                LabelAppeal
                                IMP_STARS
                                F_STARS
                                IMP_Sulphates
                                IMP_TotalSulfurDioxide
                                VolatileAcidity
                                IMP_pH
                        / dist=ZINB link=log;
        zeromodel AcidIndex
                                IMP_Alcohol
                                CitricAcid
                                IMP_FreeSulfurDioxide
                                LabelAppeal
                                IMP_ResidualSugar
```

```
                        IMP_STARS
                        F_STARS
                        IMP_Sulphates
                        IMP_TotalSulfurDioxide
                        VolatileAcidity
                        / link=logit;
                        output out=model_data pred=P_TARGET_ZINB pzero=P_ZERO_ZINB;
run;
data ZI_data;
        set model_data;
        P_ZERO_ZIP = exp(P_ZERO_ZIP) / (1.0 + exp(P_ZERO_ZIP));
        P_ZERO_ZINB = exp(P_ZERO_ZINB) / (1.0 + exp(P_ZERO_ZINB));
        P_ZERO_LOG = exp(P_ZERO_LOG) / (1.0 + exp(P_ZERO_LOG));
run;
proc sort data = ZI_data; by DESCENDING TARGET_AMT ; run;
proc print data = ZI_data (obs = 100);
        VAR TARGET P_TARGET_NB P_TARGET_POI TARGET_AMT P_TARGET_ZIP
P_TARGET_ZINB
        TARGET_FLAG P_ZERO_ZIP P_ZERO_LOG P_ZERO_ZINB;
run;
proc univariate data=ZI_data noprint;
        histogram TARGET                /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram P_TARGET_NB           /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram P_TARGET_POI          /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram TARGET_AMT            /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram P_TARGET_ZIP          /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram P_TARGET_ZINB         /midpoints = 0 1 2 3 4 5 6 7 8 9;
        histogram TARGET_FLAG;
        histogram P_ZERO_ZIP;
        histogram P_ZERO_LOG;
        histogram P_ZERO_ZINB;
run;
/*MODEL SELECTION*/
data model_out;
        set ZI_data;
        P_TARGET_ZINB               = P_TARGET_ZINB * (1 - P_ZERO_ZINB);
        P_TARGET_ZIP        = P_TARGET_ZIP * (1 - P_ZERO_ZIP);
        P_TARGET_HURDLE_NB      = P_ZERO_LOG * (P_TARGET_NB);
        P_TARGET_HURDLE_POI = P_ZERO_LOG * (1 + P_TARGET_AMT_POI);
run;

%macro FIND_ERROR( DATAFILE, P, MEANVAL );
title 'Model Error Comparison';
%let ERRFILE   = ERRFILE;
```

```
%let MEANFILE          = MEANFILE;

data &ERRFILE.;
set &DATAFILE.;
        ERROR_MEAN                  = abs( TARGET - &MEANVAL.                   )**&P.;
        ERROR_POI                   = abs( TARGET - P_TARGET_POI                )**&P.;
        ERROR_NB                    = abs( TARGET - P_TARGET_NB                 )**&P.;
        ERROR_ZIP                   = abs( TARGET - P_TARGET_ZIP                )**&P.;
        ERROR_ZINB                  = abs( TARGET - P_TARGET_ZINB               )**&P.;
        ERROR_HURDLE_NB   = abs( TARGET - P_TARGET_HURDLE_NB     )**&P.;
        ERROR_HURDLE_POI  = abs( TARGET - P_TARGET_HURDLE_POI    )**&P.;
run;

proc means data=&ERRFILE. noprint;
output out=&MEANFILE.
        mean(ERROR_MEAN)          =        ERROR_MEAN
        mean(ERROR_POI)                    =        ERROR_POI
        mean(ERROR_NB)                     =        ERROR_NB
        mean(ERROR_ZIP)                    =        ERROR_ZIP
        mean(ERROR_ZINB)          =        ERROR_ZINB
        mean(ERROR_HURDLE_NB)     =        ERROR_HURDLE_NB
        mean(ERROR_HURDLE_POI)    =        ERROR_HURDLE_POI
        ;
run;

data &MEANFILE.;
length P 8.;
set &MEANFILE.;
        P                                  = &P.;
        ERROR_MEAN                  = ERROR_MEAN                ** (1.0/&P.);
        ERROR_POI                   = ERROR_POI                 ** (1.0/&P.);
        ERROR_NB                    = ERROR_NB                  ** (1.0/&P.);
        ERROR_ZIP                   = ERROR_ZIP                 ** (1.0/&P.);
        ERROR_ZINB                  = ERROR_ZINB          ** (1.0/&P.);
        ERROR_HURDLE_NB   = ERROR_HURDLE_NB     ** (1.0/&P.);
        ERROR_HURDLE_POI  = ERROR_HURDLE_NB     ** (1.0/&P.);
        drop _TYPE_;
run;

proc print data=&MEANFILE.;
run;

%mend;
```

```
%FIND_ERROR( model_out, 1, 3.8522 ); *Average Error;
%FIND_ERROR( model_out, 1.5, 3.8522 ); *Exponent;
%FIND_ERROR( model_out, 2, 3.8522 ); *Root Mean Square Error;

/* Notes: Best Model is LOG/POI model*/
```