Homework5

John DeBlase Sekhar Mekala Sonya Hong

November 27, 2016

Project Requirements

The main goal of this project is to build several different count models that try to predict wine sales based on both marketing attributes and chemical properties of different wines. We are given 2 data sets: *training* and *test* data sets. The training data has input variables along with the observed response variable.

We will use the training data set to train our model, and the predictions obtained on the test data will be submitted as a project deliverable.

Data Exploration

The training and test data sets contain the following variables:

Figure 0: Variables for wine data set

```
##  [1] "TARGET"             "FixedAcidity"       "VolatileAcidity"
##  [4] "CitricAcid"         "ResidualSugar"      "Chlorides"
##  [7] "FreeSulfurDioxide"  "TotalSulfurDioxide" "Density"
## [10] "pH"                 "Sulphates"          "Alcohol"
## [13] "LabelAppeal"        "AcidIndex"          "STARS"
```

TARGET,the number of cases purchased, will be our response variable. According to theoretical effects from the documentation, higher numbers of stars and a nicer label suggest better sales. Other variables associated with the chemical makeup of wine do not have an obvious theoretical effect on the sale of wine.

We will first summarize and construct density plots of the predictors in the training set in order to assess distribution and look for the presence of NA values.

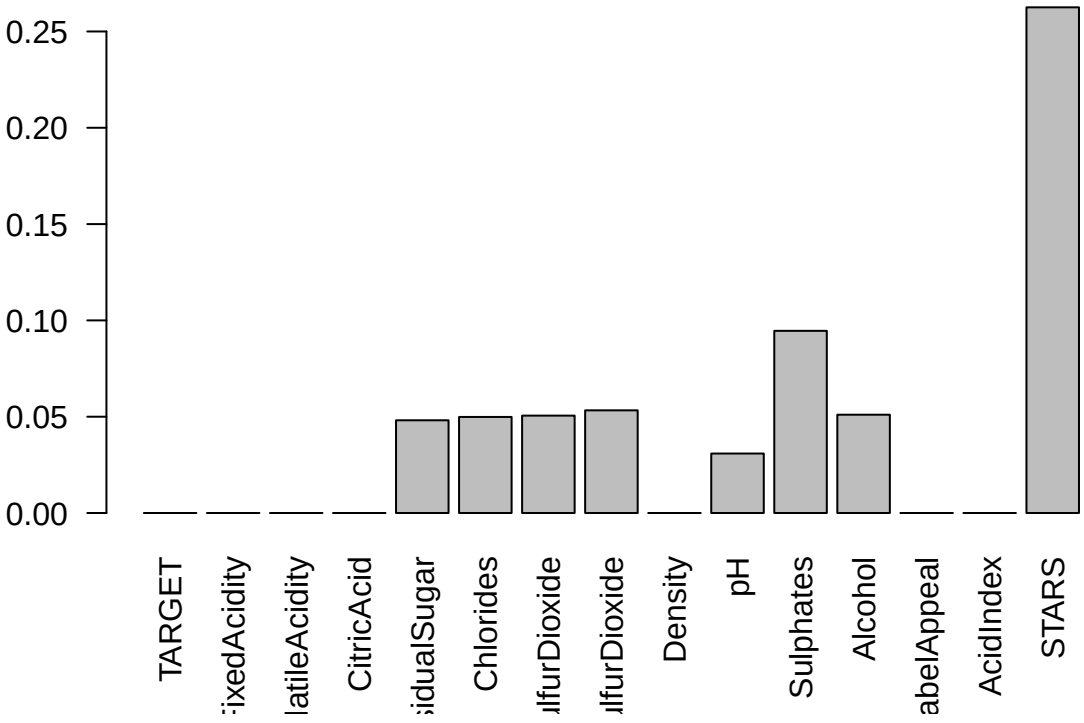
Figure 1: Summary data

```
##   FixedAcidity     VolatileAcidity     CitricAcid      ResidualSugar
##  Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400   Min.   :-127.800
##  1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300   1st Qu.:  -2.000
##  Median :  6.900   Median : 0.2800   Median : 0.3100   Median :   3.900
##  Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084   Mean   :   5.419
##  3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800   3rd Qu.:  15.900
##  Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600   Max.   : 141.150
##                                                        NA's   :616
##    Chlorides       FreeSulfurDioxide TotalSulfurDioxide    Density
##  Min.   :-1.1710   Min.   :-555.00   Min.   :-823.0     Min.   :0.8881
##  1st Qu.:-0.0310   1st Qu.:   0.00   1st Qu.:  27.0     1st Qu.:0.9877
##  Median : 0.0460   Median :  30.00   Median : 123.0     Median :0.9945
##  Mean   : 0.0548   Mean   :  30.85   Mean   : 120.7     Mean   :0.9942
##  3rd Qu.: 0.1530   3rd Qu.:  70.00   3rd Qu.: 208.0     3rd Qu.:1.0005
##  Max.   : 1.3510   Max.   : 623.00   Max.   :1057.0     Max.   :1.0992
##  NA's   :638       NA's   :647       NA's   :682
```

```
##        pH          Sulphates          Alcohol       LabelAppeal
##  Min.   :0.480   Min.   :-3.1300   Min.   :-4.70   Min.   :-2.000000
##  1st Qu.:2.960   1st Qu.: 0.2800   1st Qu.: 9.00   1st Qu.:-1.000000
##  Median :3.200   Median : 0.5000   Median :10.40   Median : 0.000000
##  Mean   :3.208   Mean   : 0.5271   Mean   :10.49   Mean   :-0.009066
##  3rd Qu.:3.470   3rd Qu.: 0.8600   3rd Qu.:12.40   3rd Qu.: 1.000000
##  Max.   :6.130   Max.   : 4.2400   Max.   :26.50   Max.   : 2.000000
##  NA's   :395     NA's   :1210      NA's   :653
##    AcidIndex          STARS
##  Min.   : 4.000   Min.   :1.000
##  1st Qu.: 7.000   1st Qu.:1.000
##  Median : 8.000   Median :2.000
##  Mean   : 7.773   Mean   :2.042
##  3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   :17.000   Max.   :4.000
##                   NA's   :3359
```

Based on the above summary data and density plots found in **Appendix A**, the predictors appear to be normally distributed. There is a moderate to high presence of NA values in several of the chemical categories and STARS categories.

Figure 2: Percentage of NA values



The variables Chlorides, FreeSulfurDioxide, TotalSulfurDioxide, pH, Sulphates and Alcohol have NA values that are less than 10% of the total observations. These continuous missing values can be imputed using a predictive mean matching algorithm from the mice package.

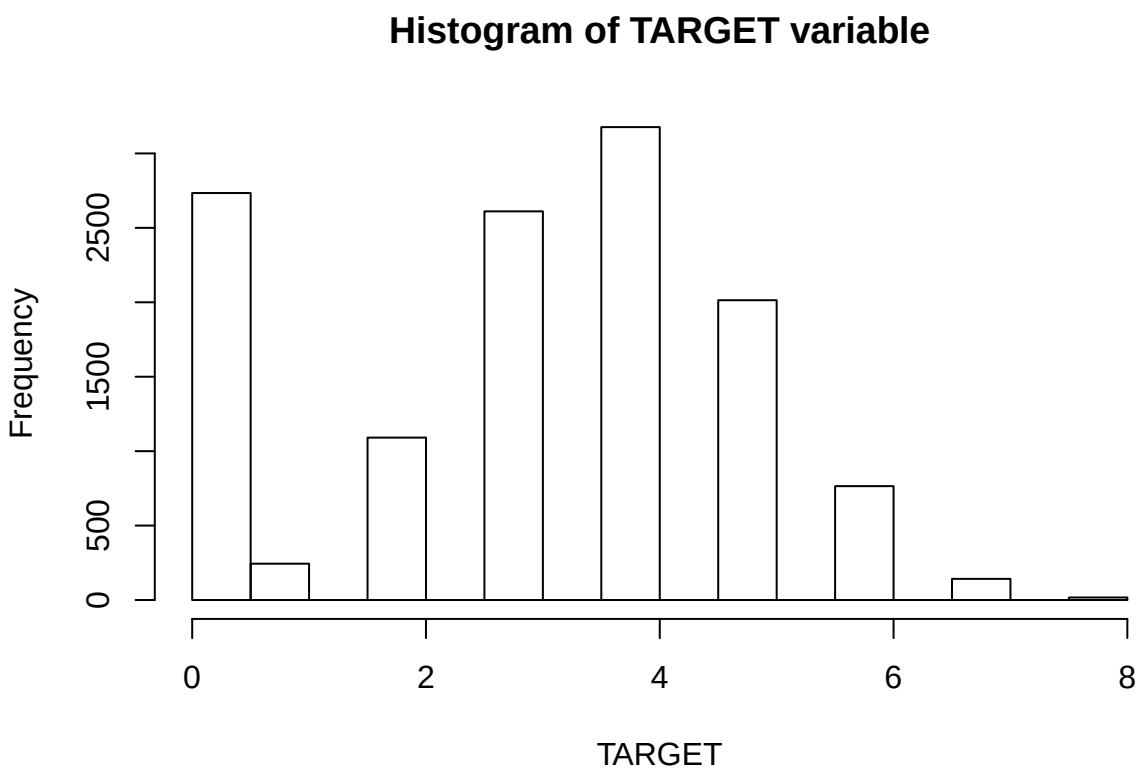
About 26% of observations for the STARS variable contains an NA value. Due to STARS being regarded as a significant predictor based on the information in the above table, we will not drop the variable from the dataset, but instead create a dummy variable to flag NA values and impute missing values in the original based on median.

Below is a summary and histogram of the distribution of the target variable.

Figure 3: Summary of the TARGET variable

```
##      TARGET
##  Min.   :0.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :3.029
##  3rd Qu.:4.000
##  Max.   :8.000
```

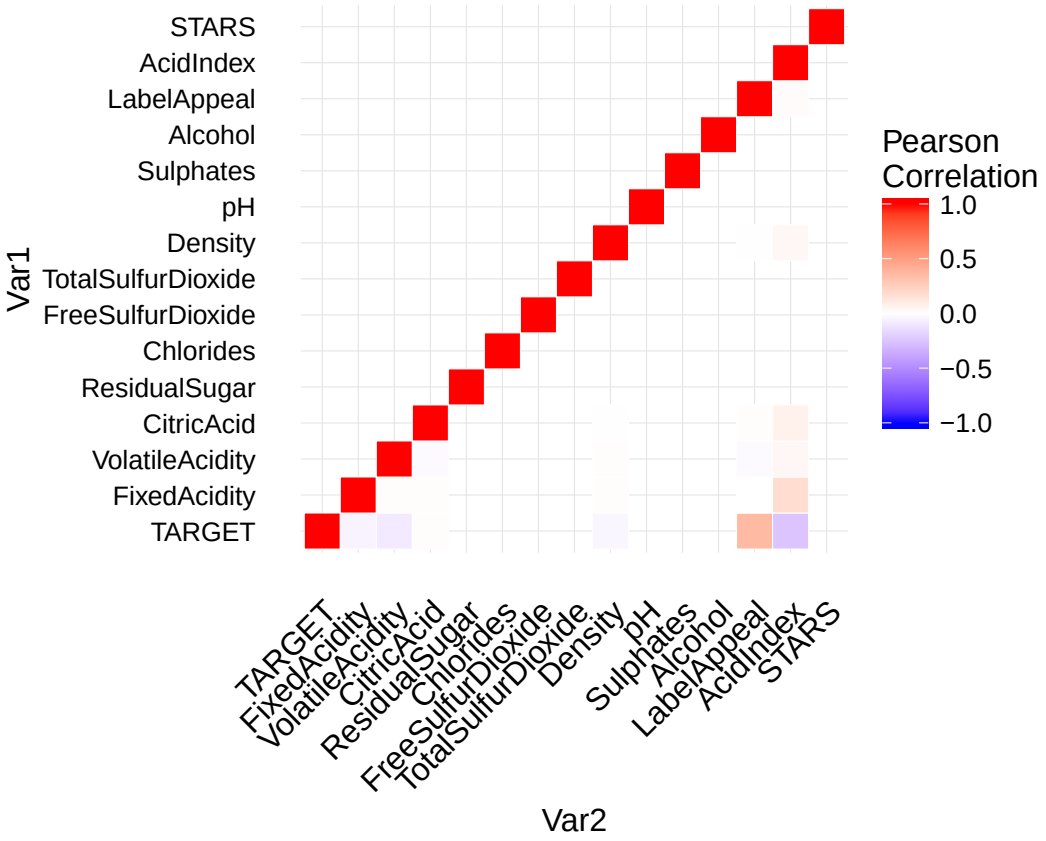
Figure 4: Histogram of the TARGET variable



The histogram and summary of our TARGET variable shows a normal distribution with a mean of 3 between purchases of 1 and 8 cases. The variable does contain an excessive amount of zeros. This is a good indicator that a zero-inflated model that takes into consideration both binomial and count regression for the TARGET variable might be appropriate for this dataset.

Finally, we will use a heatmap to look for any pronounced correlations between the target variable and predictors.

Figure 5: Heatmap of training dataset



The heatmap shows that there is little to no correlation amongst many of the chemical variables. We can see a slight negative correlation for AcidIndex and TARGET, as well as a slightly positive correlation between LabelAppeal and TARGET.

We will assess the correlations once again after missing variables have been imputed and the STARS dummy variable is created in the next section to see if these changes significantly impact our heatmap results in any way.

Assesment for Data Prepartion

- Impute all continuous variables with mice. Impute categorical variable STARS with median. Create dummy variable for STARS to indicate NA values in the dataset.
- The dataset overall is highly un-correlated with normally distributed continuous variables. Only slight correlations are present with AcidIndex, LabelAppeal and possiblty STARS. The relationships of these variables will most likely be the primary focus of the investigation.
- The TARGET count has an excess number of zeros but is otherwise normally distributed with a mean of 3. The count regression models will show which predictors will affect this number.
- The excess zeros indicate that zero-inflated models could be more relevant in the model building for the count-regression. Zero-inflated models will be built by seperating TARGET=0 from TARGET>0 and creating a TARGET CLASS variable(0,1). A binomial logistic regression will be fitted to TARGET CLASS alongside the count data in TARGET using Poisson and Negative Binomial models for TARGET>1. The zeroinfl() R function will also be used to create a singular zero-inflated model for comparison and evaluation.

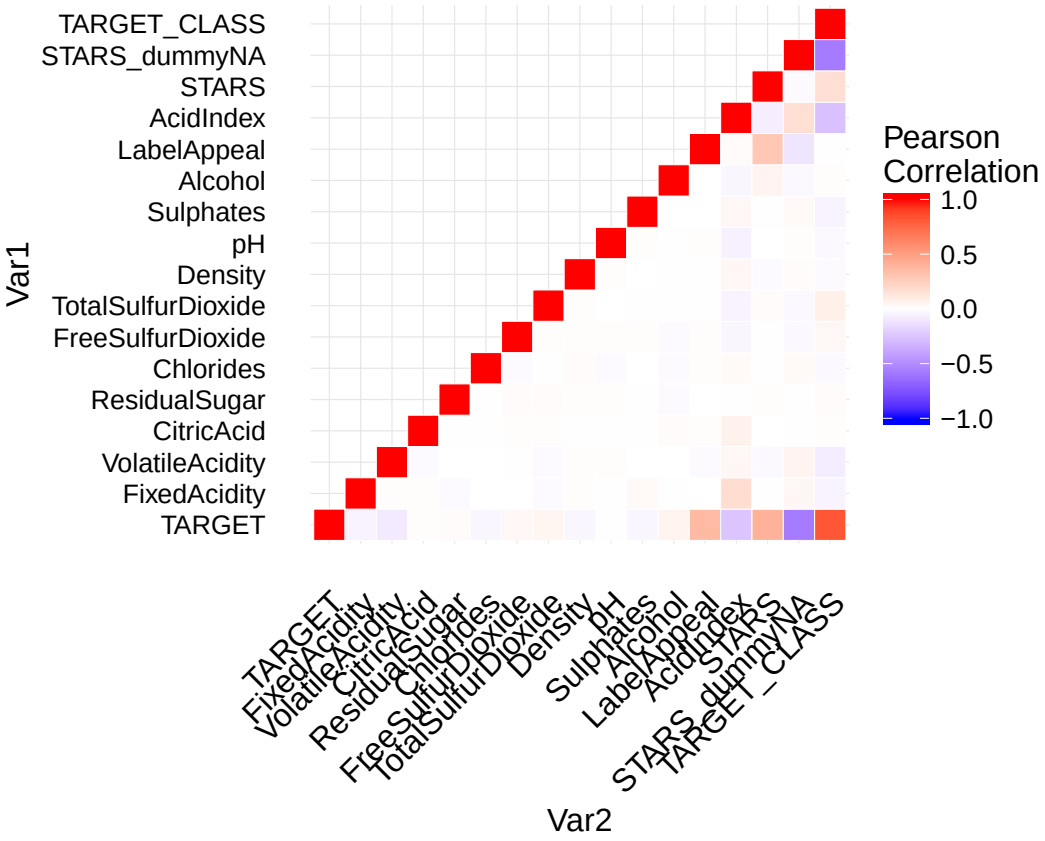
Data Preparation

After imputation and dummy variable creation, the modified set of variables for the training set is given below. Similar transformations will be made to the test set before any model is used for prediction.

```
##  [1] "TARGET"             "FixedAcidity"       "VolatileAcidity"
##  [4] "CitricAcid"         "ResidualSugar"      "Chlorides"
##  [7] "FreeSulfurDioxide"  "TotalSulfurDioxide" "Density"
## [10] "pH"                 "Sulphates"          "Alcohol"
## [13] "LabelAppeal"        "AcidIndex"          "STARS"
## [16] "STARS_dummyNA"      "TARGET_CLASS"
```

We want to check the dataset for correlations now that data has been imputed and new variables added.

Figure 6: Heatmap after imputation and dummy variables added



The heatmap now shows a strong positive correlation between STARS and the TARGET. There is also a strong negative correlation between an NA value in STARS and LabelAppeal as well as a positive correlation with AcidIndex. The new TARGET_CLASS variable shows a strong negative correlation with missing STARS values and AcidIndex and a positive correlation with STARS.

Assessment for Model Building and Selection

- Based on the results of the heatmap, particular attention will be paid to STARS, STARS_dummyNA, LabelAppeal and AcidIndex during model evaluation.

- Both normal and zero-inflated poisson models will be built using stepwise forward and backward selection. Both normal and zero-inflated negative binomial models will be created using the same process. A binary logistic model alongside count models on subsetted training data will also be created.
- In addition, two Multiple Linear Regression models will be built for comparison with the count models.
- Results for all models will be Cross-validated with the cv.glm() function for model selection.
- We will test for overdispersion and build negative binomial models using the *disperionTest()* function from the pscl package. If the overdispersion value is statistically significant, this will indicate whether a negative binomial model is more appropriate than a poisson model for this data.
- If the binary logistic regression model is chosen, a confusion matrix and ROC curve will be calculated to evaluate model accuracy.

Model Building

Poisson Models

Poisson-Model-1

We will first build and interpret several different poisson models. Our first model will be a regular poisson model that does not take the excess zeros into account (that is, treats the 0 as normal count). We will use the *stepAIC()* function to select variables using the forward and backward stepwise methods.

The coefficients for the first model (*Poisson-Model-1*) are given below. Note that the variables shown in the model are the significant variables identified by *stepAIC()* function of R, using forward and backward stepwise variable selection method:

Figure 7: Regular Poisson Model coefficients with stepwise selection

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	1.7833686	0.1953495	9.129120	0.0000000
VolatileAcidity	-0.0311900	0.0065169	-4.785992	0.0000017
Chlorides	-0.0374619	0.0160820	-2.329426	0.0198365
FreeSulfurDioxide	0.0001012	0.0000342	2.960357	0.0030728
TotalSulfurDioxide	0.0000785	0.0000221	3.554089	0.0003793
Density	-0.2785180	0.1917921	-1.452187	0.1464496
pH	-0.0137160	0.0075316	-1.821118	0.0685890
Sulphates	-0.0111819	0.0055020	-2.032337	0.0421195
Alcohol	0.0028939	0.0013740	2.106165	0.0351900
LabelAppeal	0.1589580	0.0061269	25.944191	0.0000000
AcidIndex	-0.0805919	0.0045161	-17.845480	0.0000000
STARS	0.1879367	0.0060916	30.851735	0.0000000
STARS_dummyNA	-1.0233056	0.0169812	-60.261057	0.0000000

The interpretation of the above model is given below:

- For one unit increase in VolatileAcidity, the average TARGET variable decreases by approximately 3%.
- For one unit increase in Chlorides, the average TARGET variable decreases by approximately 3.7%.
- FreeSulfurDioxide and TotalSulfurDioxide have negligible impact on the TARGET variable, since their coefficinents are almost 0.

- A one unit increase in Density variable, will make the average TARGET variable decrease by 27.8%. Although the Density variable's p-value is not significant (assuming the significance level as 5%), we still consider the Density variable as significant, since this variable is chosen by the variable selection method (which is insensitive to the Type-1 errors).
- A one unit increase in pH value will decrease the average value of TARGET by 1%. Same is applicable to Sulphates also.
- A one unit increase in Alcohol variable will increase the TARGET variable's average value by just 0.2%
- The LabelAppeal and STARS have a positive impact on the average TARGET value. A one unit increase in LabelAppeal will increase the TARGET variable by 15.9%, and a one unit increase in STARS will increase the average TARGET value by 18.8%.
- The dummy variable STARS_dummyNA has a negative impact on the average TARGET value. Whenever the STARS variable has unavailable information, the average TARGET value will decrease by 102%, and this means, the TARGET value will become zero, with one unit increase in STARS_dummyNA variable. But the STARS_dummyNA will only take 0 or 1 values, and hence there is no possibility for this variable to increase more than 1 unit from 0 value. This is an example, where extrapolation of the model on out of range data will give unpredictable results.

Poisson-Model-2:

Our second model will be a "Hurdle" or two part model which is actually composed of two models. The first model will use binomial logistic regression to predict if the TARGET variable is zero. Given that our first model predicts the TARGET variable as non-zero, we will then use Poisson regression to predict the TARGET value, which will be greater than zero. This approach will help us eliminate the effect of the bloated zero values in the TARGET variable. For the classification, we will use logistic regression. If the logistic regression model does not give better results (based on accuracy), we will consider other classification models. For both the models (logistic and poisson) we will use *stepAIC()* to determine the significant variables.

The binary logistic regression model and count data model on the subsetted training data are given below (these models are obtained after eliminating insignificant variables using *stepAIC()* method):

NOTE: For logistic regression purposes, let TARGET_CLASS be a variable that identifies if the TARGET variable's value is greater than 0. If TARGET > 0, then TARGET_CLASS will be 1, else TARGET_CLASS will be zero. With this assumption, a logistic model is fit to determine the TARGET_CLASS value.

Figure 8: Binomial Logistic Regression on TARGET CLASS

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	2.7074215	0.2789457	9.705907	0.0000000
VolatileAcidity	-0.1827813	0.0365013	-5.007533	0.0000006
Chlorides	-0.1627533	0.0900704	-1.806956	0.0707692
FreeSulfurDioxide	0.0006380	0.0001948	3.274738	0.0010576
TotalSulfurDioxide	0.0008199	0.0001234	6.643789	0.0000000
pH	-0.1887531	0.0420467	-4.489127	0.0000072
Sulphates	-0.1020094	0.0305937	-3.334324	0.0008551
Alcohol	-0.0233085	0.0076892	-3.031317	0.0024349
LabelAppeal	-0.4670267	0.0333244	-14.014576	0.0000000
AcidIndex	-0.3888462	0.0213956	-18.174125	0.0000000
STARS	2.5594271	0.1119028	22.871867	0.0000000
STARS_dummyNA	-4.3759446	0.1114705	-39.256526	0.0000000

As per the above model, as the STARS value increases, so does the probability that TARGET value is more than 0, since STARS has a large positive coefficient (2.56), when compared to other variables coefficients. As the STARS_dummyNA value increases, so does the probability that the TARGET variable will be 0, since STARS_dummyNA has a big negative coefficient (-4.38) when compared to other variables coefficients. For all other variables if the coefficient is positive then the probability of TARGET>0 increases, and if the coefficient is negative, then the probability that TARGET = 0 increases.

Figure 9: Poisson using a subset of training data (TARGET >= 1)

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	1.1977696	0.0416754	28.740416	0.0000000
VolatileAcidity	-0.0098220	0.0065535	-1.498755	0.1339373
Alcohol	0.0064371	0.0013717	4.692801	0.0000027
LabelAppeal	0.2180374	0.0061855	35.249521	0.0000000
AcidIndex	-0.0154504	0.0046618	-3.314268	0.0009188
STARS	0.0923608	0.0062616	14.750307	0.0000000
STARS_dummyNA	-0.1442585	0.0170476	-8.462082	0.0000000

The poisson regression model using only subset data (TARGET>0) is displayed in Figure 9. These results are applicable to scenarios where the TARGET variable is predicted to be greater than 0. The model coefficients can be interpreted as given below:

- For one unit increase in VolatileAcidity, the TARGET value decreases by 0.9%, which is not significant.
- For one unit increase in Alcohol, the TARGET value increases by 0.64%, which is not significant.
- For one unit increase in LabelAppeal, the TARGET value increases by 21.8%, which is significant.
- For one unit increase in AcidIndex, the TARGET value decreases by 1.5%, which is not significant.
- For one unit increase in STARS, the TARGET value increases by 9%, which is significant.
- For one unit increase in STARS_dummyNA, the TARGET value decreases by 14.42%, which is significant.

Poisson-Model-3:

In *Poisson-Model-2* we used a two part method to determine if the TARGET value is greater than 0, and then applied another model (poisson regression), given that the TARGET value is estimated to be greater than 0. In that approach, we assumed that the 0 value of the TARGET variable is generated by a separate single process, and the TARGET variable of greater than 1 is generated by another separate process. These two processes are assumed to be mutually exclusive. That is, the second process will never generate 0 value for TARGET, and the first process will never generate non-zero value for TARGET.

However, there might be a loss in prediction performance if zeros are generated by both the processes. So zero inflated models consider that there is a chance that the second process can generate a value of TARGET equal to 0. We name the zero inflated model produced using logistic regression and poisson regresion as *Poisson-Model-3*. We will use *zeroinfl()* function to produce a two component zero inflated poisson model via maximum likelihood. Since stepAIC does not work specifically for the output of this function, we will use the variables of logistic regression and poisson regression obtained in *Poisson-Model-2*.

The third zero-inflated poisson model coefficients are given below:

Figure 10: Coefficients for logit results from zeroinfl()

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	-2.2359646	0.4533581	-4.932006	0.0000008
VolatileAcidity	0.1848654	0.0432196	4.277350	0.0000189
Chlorides	0.1280032	0.1058891	1.208842	0.2267234
FreeSulfurDioxide	-0.0007880	0.0002322	-3.393837	0.0006892
TotalSulfurDioxide	-0.0009203	0.0001463	-6.289569	0.0000000
pH	0.2114782	0.0497176	4.253593	0.0000210
Sulphates	0.1230826	0.0363175	3.389072	0.0007013
Alcohol	0.0305835	0.0091652	3.336934	0.0008471
LabelAppeal	0.7216685	0.0423219	17.051879	0.0000000
AcidIndex	0.4282751	0.0255731	16.747125	0.0000000
STARS	-3.8139139	0.3305249	-11.538961	0.0000000
STARS_dummyNA	5.8790495	0.3308921	17.767273	0.0000000

The *zeroinfl()* function of R, uses a logistic model based on calculating the TARGET = 0. In *Poison-model-2*, the logistic regression was developed based on the probability of TARGET>0. Hence the coefficients signs of logistic models in *Poisson-model-2* and *Poisson-model-3* are flipped. For instance the STARS variable has a positive coefficient in the logistic model of *Poisson-model-2*, while the STARS variable in *Poisson-model-3* has a negative sign.

Let us now interpret the logistic model of *Poisson-model-3*.

- The STARS_dummyNA variable has a big positive coefficient. Hence as this variable increases the probability of TARGET = 0 increases significantly. For STARS variable we have a big negative coefficient. This will make the probability of TARGET=0 decrease significantly as STARS variable increases. On the similar lines if any variable has a positive coefficient, the probability of TARGET = 0 will increase, with the increase in the variable value. If the variable has negative coefficient, then an increase in the variable value will decrease the probability of TARGET=0.

Figure 11: Coefficents for Count model from zeroinfl()

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	1.1793525	0.0429439	27.462625	0.0000000
VolatileAcidity	-0.0124677	0.0067045	-1.859598	0.0629425
Alcohol	0.0066713	0.0014046	4.749437	0.0000020
LabelAppeal	0.2319512	0.0063152	36.729238	0.0000000
AcidIndex	-0.0196370	0.0048246	-4.070153	0.0000470
STARS	0.1046751	0.0064036	16.346280	0.0000000
STARS_dummyNA	-0.1827094	0.0185694	-9.839297	0.0000000

Figure 11 shows the poisson regression of *Poisson-model-3*. This model interpretation is given below:

- For one unit increase in LabelAppeal, the average TARGET value will increase by 23.2% approximately. The STARS variable also has same positive impact, but it will increase the TARGET value by 10.5%. The Alcohol variable has positive impact, but a unit increase in Alcohol will only increase the TARGET variable by 0.7% approximately.
- All other variables in the model have negative impact on the TARGET variable. One unit increase in STARS_dummyNA variable will make TARGET variable decrease by 18.3% approximately. A one unit increase in VolatileAcidity will reduce the TARGET value by 1.2%, and one unit increase in AcidIndex will make the TARGET variable decrease by 1.96%

NOTE: The above variables interpretation is applicable to scenarios where the TARGET variable is predicted to be more than 0.

We will now check the pure poisson models for over-dispersion using *dispersiontest()* from the AER package. This will help us gauge the validity of any negative binomial models.

```
##
##  Overdispersion test
##
## data:  pois1
## z = -8.6234, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.8876538
```

```
##
##  Overdispersion test
##
## data:  zero_pois1
## z = -247.03, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.1798734
```

P-values of 1 for both tests show that over-dispersion is not present in the poisson models. We can therefore conclude that Negative binomial models might not be appropriate, since there is no over dispersion in the poisson models. However we will develop negative binomial models to determine if these models improve the prediction accuracy any further.

Negative Binomial Models

We will build and interpret our negative binomial models in a manner similar to the poisson models using the *glm.nb()* function from the MASS package. First the stepAIC function will perform forward and backward selection to produce a regular negative binomial model that does not account for zeros.

Negative-binomial-model-1

The coefficients for our first negative binomial model are shown below. This model does not consider the zero inflation, and treats zeros in the TARGET variable as a normal value:

Figure 12: Results of regular negative binomial regression using stepAIC

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	1.7811394	0.1953831	9.1161407	0.0000000
FixedAcidity	0.0000122	0.0008196	0.0149198	0.9880962
VolatileAcidity	-0.0310278	0.0065182	-4.7601888	0.0000019
CitricAcid	0.0055649	0.0058951	0.9439808	0.3451795
ResidualSugar	0.0001172	0.0001514	0.7743720	0.4387108
Chlorides	-0.0371935	0.0160851	-2.3123021	0.0207610
FreeSulfurDioxide	0.0001006	0.0000342	2.9432408	0.0032480

	Coefficient	Std_Error	Z_Score	P_value
TotalSulfurDioxide	0.0000781	0.0000221	3.5367357	0.0004051
Density	-0.2759467	0.1918310	-1.4384888	0.1502954
pH	-0.0138824	0.0075340	-1.8426227	0.0653841
Sulphates	-0.0110826	0.0055032	-2.0138507	0.0440252
Alcohol	0.0028822	0.0013746	2.0966784	0.0360220
LabelAppeal	0.1589000	0.0061275	25.9324448	0.0000000
AcidIndex	-0.0808696	0.0045723	-17.6869503	0.0000000
STARS	0.1878865	0.0060928	30.8376475	0.0000000
STARS_dummyNA	-1.0229932	0.0169841	-60.2323709	0.0000000

Let us interpret the above model in figure-12:

- The variables FixedAcidity, CitricAcid, ResidualSugar, FreeSulfurDioxide, TotalSulfurDioxide, Alcohol, LabelAppeal, and STARS have positive coefficients. But except the STARS and LabelAppeal, all other listed variable do not have a significant positive coefficients. A unit increase in STARS will increase the TARGET value by 18.8%, and a one unit increase in LabelAppeal will increase the TARGET variable by approximately 16%. Other variables FixedAcidity, CitricAcid, ResidualSugar, FreeSulfurDioxide, TotalSulfurDioxide, and Alcohol have positive effect on TARGET variable, although the effect is less than 1% increase in TARGET variable.
- The variables VolatileAcidity, Chlorides, Density, pH, Sulphates, AcidIndex and STARS_dummyNA have negative coefficients. But except the Density and STARS_dummyNA variables, all other listed variable do not have a significant negative coefficients. A unit increase in Density will decrease the TARGET value by 27.6%, and a one unit decrease in STARS_dummyNA will decrease the TARGET variable by approximately 10%. Other variables VolatileAcidity, Chlorides, pH, Sulphates, and AcidIndex have negative effect on TARGET variable, although the effect is less than 1% decrease in TARGET variable.

The results of this model are the same as the one we obtained using poisson regression (*Poisson-model-1*), and this confirms that there is no overdispersion in the data.

Negative-binomial-model-2

Our second model will be a Hurdle or two part model. The first part of the model will predict if the TARGET variable is equal to zero. Given that the first part predicts the TARGET >0, the second part predicts the TARGET value using Negative binomial model. We will not develop a separate logistic model (part 1), since that model was already developed while developing the *Poisson-model-2*. The second part of the model uses negative binomial distribution trained on the data for which the TARGET > 0. The model is further refined using *stepAIC()* to eliminate insignificant variables.

The coefficients of our second negative binomial model using a subset of the training set (TARGET>0) is given below:

Figure 13: Negative binomial using a subset of training data (TARGET >= 1)

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	1.1977695	0.0416757	28.740248	0.0000000
VolatileAcidity	-0.0098220	0.0065535	-1.498747	0.1339393
Alcohol	0.0064371	0.0013717	4.692777	0.0000027
LabelAppeal	0.2180375	0.0061856	35.249331	0.0000000
AcidIndex	-0.0154504	0.0046618	-3.314251	0.0009189

	Coefficient	Std_Error	Z_Score	P_value
STARS	0.0923608	0.0062617	14.750220	0.0000000
STARS_dummyNA	-0.1442587	0.0170477	-8.462048	0.0000000

We eliminated the TARGET=0 records from the training data and fit a negative binomial model. The *stepAIC()* function has eliminated all the unnecessary variables and only 6 variables are included in the final negative binomial model. The coefficients of this model (given in Figure 13) are interpreted below:

- A one unit increase in LabelAppeal will increase the TARGET variable value by 22%.
- A one unit increase in STARS will increase the TARGET value by 9.2%
- A one unit increase in Alcohol will increase the TARGET value by 0.6%
- A one unit increase in VolatileAcidity will decrease the TARGET value by 1%
- A one unit increase in AcidIndex will decrease the TARGET variable by 1.5%
- A one unit increase in STARS_dummyNA will decrease the TARGET value by 14.4%

Again, this model's coefficients are the same as the model obtained in *Poisson-model-2* (developed as a 2 part model with logistic regression to predict TARGET > 0 and poisson regression model to predict TARGET, given that TARGET is predicted as more than 0. See Figure-9 for *Poisson-model-2*). Since the poisson and negative binomial models resulted the same models, we can conclude that there is no overdispersion in the data.

Negative-binomial-model-3

Now we will develop a zero inflated model using negative binomial distribution and logistic regression. The main difference between this model (*Negative-binomial-model-3*) and the previous model(*Negative-binomial-model-2*) is related to which process generates the TARGET=0 values. In the hurdle model (*Negative-binomial-model-2*), the TARGET=0 values are generated by a single process, and another process generates TARGET > 0 values. These two processes are assumed to generate non-overlapping values. But in zeroinflated models, the TARGET=0 values are generated by both the processes, although the first process still generates TARGET=0 values only. We will use *zeroinfl()* function of R to fit a logistic regression model (to predict if TARGET=0), and negative binomial model to predict the TARGET value, given that the logistic regression model predicts that the TARGET value is generated by the second process.

The results from *Negative-binomial-model-2* model will be used as predictors for the two components (logistic regression and negative binomial regression model).

The coefficients of our third negative binomial model are shown below:

Figure 14: Coefficients for logit model from zeroinfl() negative binomial

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	-2.2358011	0.4533971	-4.931221	0.0000008
VolatileAcidity	0.1848577	0.0432195	4.277188	0.0000189
Chlorides	0.1280146	0.1058889	1.208952	0.2266814
FreeSulfurDioxide	-0.0007881	0.0002322	-3.394416	0.0006878
TotalSulfurDioxide	-0.0009202	0.0001463	-6.288391	0.0000000
pH	0.2114853	0.0497174	4.253745	0.0000210
Sulphates	0.1230844	0.0363174	3.389134	0.0007011
Alcohol	0.0305846	0.0091651	3.337066	0.0008467

	Coefficient	Std_Error	Z_Score	P_value
LabelAppeal	0.7216679	0.0423218	17.051925	0.0000000
AcidIndex	0.4282764	0.0255731	16.747153	0.0000000
STARS	-3.8141132	0.3305782	-11.537702	0.0000000
STARS_dummyNA	5.8792318	0.3309455	17.764955	0.0000000

The logistic regression model produced (above in figure 14) by the zero inflation method is same as the logistic model produced by *Poisson-model-3*. See the logistic model in *Poisson-model-3* (Figure-10) for interpretation of the above logistic model.

Figure 15: Coefficients for count model from zeroinfl() negative binomial

	Estimate	Std..Error	z.value	Pr. . . z..
(Intercept)	1.1793519	0.0429439	27.462642	0.0000000
VolatileAcidity	-0.0124692	0.0067045	-1.859831	0.0629095
Alcohol	0.0066714	0.0014046	4.749564	0.0000020
LabelAppeal	0.2319500	0.0063152	36.729088	0.0000000
AcidIndex	-0.0196368	0.0048246	-4.070114	0.0000470
STARS	0.1046750	0.0064036	16.346279	0.0000000
STARS_dummyNA	-0.1827106	0.0185693	-9.839372	0.0000000

The model displayed above is the same regression model (based on zero inflation poisson regression) obtained in *Poisson-model-3* model (Figure-11).

We can see that the poisson models and negative binomial models are the same. This is due to the fact that there is no over disperssion for the poisson models. From now onwards we will drop the negative binomial models and consider only the poisson models _Poisson-model-1, *Poisson-model-2* and *Poisson-model-3*.

Multivariate Linear Regression

Now we will fit multi variate regression models, one linear regression and one polynomial regression with optimal degree, selected using a cross-validation method.

Linear-model-1

Our first linear regression model will be built again using forward and backward stepwise selection using stepAIC.

The coefficients of the first MLR (Multivariate Linear Regression) model are shown below:

Figure 16: Coefficients for MLR using stepwise selection

	Coefficient	Std_Error	Z_Score	P_value
(Intercept)	4.3924733	0.4441574	9.889453	0.0000000
VolatileAcidity	-0.0963434	0.0148168	-6.502302	0.0000000
Chlorides	-0.1191136	0.0365165	-3.261913	0.0011095
FreeSulfurDioxide	0.0002847	0.0000780	3.651370	0.0002619
TotalSulfurDioxide	0.0002230	0.0000499	4.463510	0.0000081
Density	-0.7963078	0.4371094	-1.821758	0.0685150

	Coefficient	Std_Error	Z_Score	P_value
pH	-0.0347089	0.0170935	-2.030530	0.0423233
Alcohol	0.0106345	0.0031217	3.406678	0.0006596
LabelAppeal	0.4665742	0.0136706	34.129805	0.0000000
AcidIndex	-0.2006850	0.0089629	-22.390570	0.0000000
STARS	0.7798366	0.0156768	49.744537	0.0000000
STARS_dummyNA	-2.2452901	0.0269499	-83.313579	0.0000000

The above model (figure 16) is obtained by fitting a linear model using all the variables of the training data, and eliminating the insignificant variables using *stepAIC()* function. The model interpretation is given below:

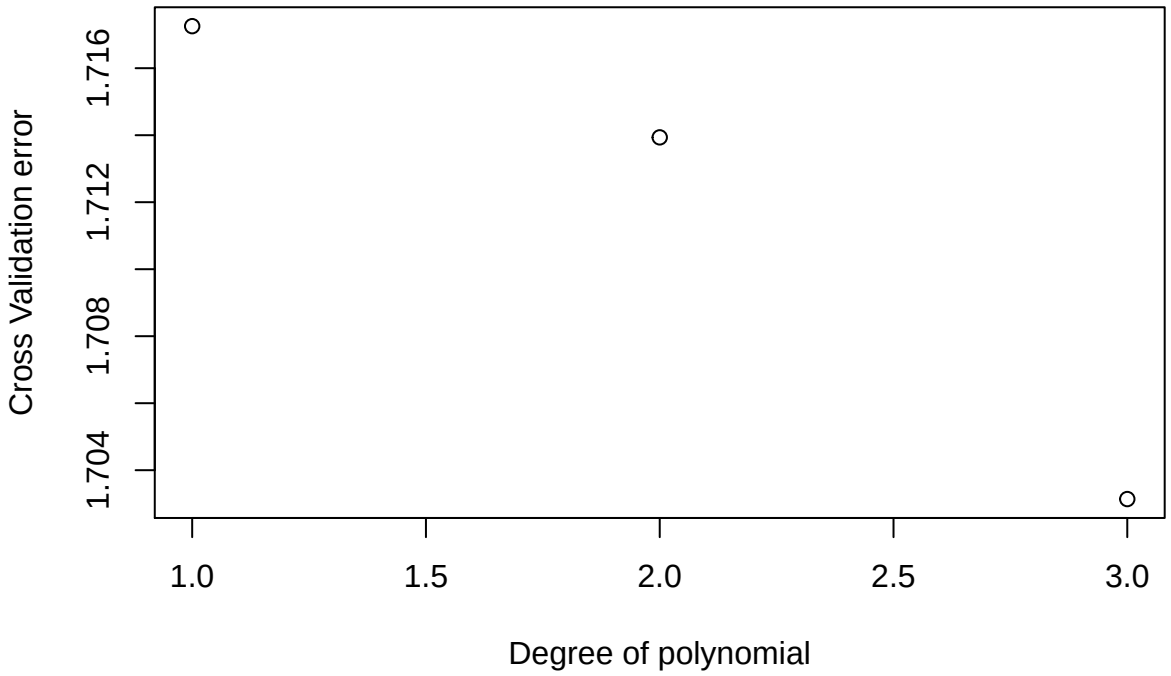
- For a unit increase in VolatileAcidity value, the TARGET variable is reduced by approximately 0.096
- For a unit increase in Chlorides value, the TARGET variable is reduced by approximately 0.12
- For a unit increase in Density value, the TARGET variable is reduced by approximately 0.8
- For a unit increase in pH value, the TARGET variable is reduced by approximately 0.035
- For a unit increase in AcidIndex value, the TARGET variable is reduced by approximately 0.2
- For a unit increase in STARS_dummyNA value, the TARGET variable is reduced by approximately 2.25 (A significant decrease)
- For a unit increase in FreeSulfurDioxide value, the TARGET variable is increased by approximately 0.00028
- For a unit increase in TotalSulfurDioxide value, the TARGET variable is increased by approximately 0.00022
- For a unit increase in Alcohol value, the TARGET variable is increased by approximately 0.01
- For a unit increase in LabelAppeal value, the TARGET variable is increased by approximately 0.47
- For a unit increase in STARS value, the TARGET variable is increased by approximately 0.7

The p-value associated with the F-Statistic of *Linear-model-1* is almost 0, which shows that there is an association between the TARGET and the independent variables in the *Linear-model-1*.

Linear-model-2

Now we will fit a polynomial regression model to the training data. But we will pick the optimal degree of polynomial using the cross validation technique. Also the polynomial model is fit only on the variables used in the *Linear-model-1*. We used polynomial regression from degree 1 to degree 3, and obtained the cross validation results (plotted in Figure-17). The figure shows that there is no significant decrease in the CV error (see the y-axis values. We are seeing only a slight decrease in the error).

Figure 17: Finding the optimal polynomial degree based on the Cross Validation technique



Since there is no significant error drop with the higher degree of polynomial. Therefore we will discard *Linear-model-2* from further consideration. Although we may only select a count model for this assignment, *Linear-model-1* will be analyzed regardless in the model selection process.

Model Selection

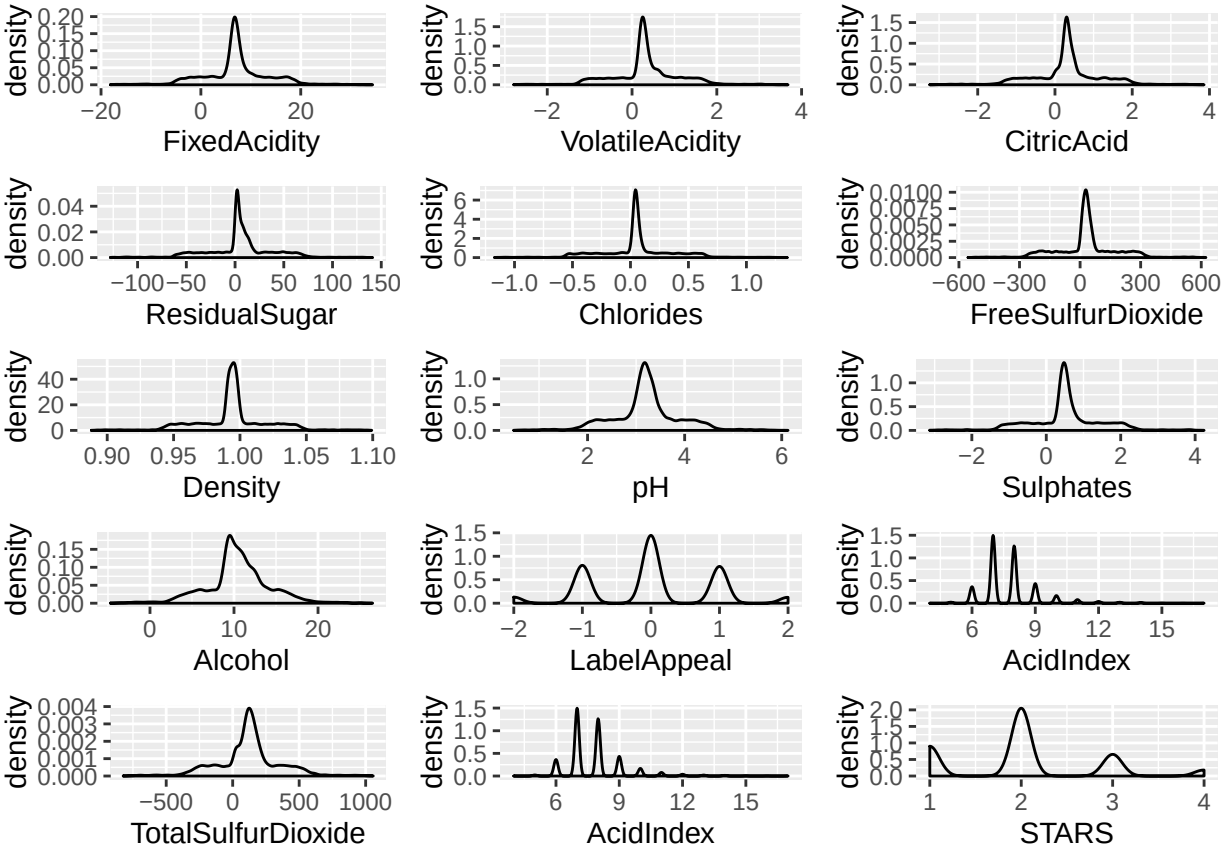
We have 12795 observations in the training data set. We will randomly select 1000 observations from this data set, and keep them aside as the test data set to evaluate the performance of our models. The remaining 11795 observations will be used to develop *Poisson-model-1* (Plain poisson model), *Poisson-model-2* (Hurdle or two part model), *Poisson-model-3* (Zero inflated poisson model) and *Linear-model-1* (Linear regression model), and their performance is evaluated on the held data. This is repeated 10 times, and the average error is calculated. The method that has the least error will be finally proposed.

Models	Error
Plain Poisson	1.380580
Hurdle Poisson	1.470034
Zero infl Poisson	1.284569
Linear model	1.389964

Since the zero-inflated poisson model has the least error, we propose the zero inflated poisson model as the final model. We will use this model to make predictions. The code for predictions can be found in the **Code Appendix** and results are saved to *evaluations_results.csv*

Appendix A: Density Plots

Density plots showing distributions of variables in the dataset



Code Appendix

Prediction Code

```
# Poisson-model-3:
zero_pois2 = zeroinfl( TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA |
   VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA
        ,data=train2, link="logit", dist="poisson")


#Generating the evaluation data predictions using Poisson-model-3

df <- read.csv("wine-evaluation-data.csv")
#names(df)
df$TARGET = round(predict(zero_pois2,newdata=test_df,type="response"))
#head(df)
write.csv(df,file="evaluation_results.csv",row.names=FALSE)
```

Analysis Code

```
library(ggplot2)
library(reshape2)
library(MASS)
library(pscl)
library(AER)
library(mice)
library(lattice)
library(knitr)
library(boot)

## utility functions

check_NA = function(df){
    apply(df,2,function(x){ sum(is.na(x))/length(x) })
}

make_cor_heatmap = function(mydata){
    # based on open source code found here:
    #http://www.sthda.com/english/wiki/
    #ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization

    cormat = round(cor(mydata),2)

    get_upper_tri = function(cormat){
        cormat[lower.tri(cormat)]<- NA
        return(cormat)
    }
```

```
    melted_cormat = melt(get_upper_tri(cormat), na.rm =T) # melt matrix

    # return ggplot object
    ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
        geom_tile(color = "white")+
        scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                             midpoint = 0, limit = c(-1,1), space = "Lab",
                             name="Pearson\nCorrelation") +
        theme_minimal()+
        theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                         size = 12, hjust = 1))+
        coord_fixed()
}

make_density_plots = function(df){

    apply(df,2,function(col){
        densityplot(col,xlab=colnames(col))
    })
}


# LOAD IN ..
setwd("~/Documents/CUNY/Data_Mining_621/Homework5")
train = read.csv("wine-training-data.csv")
train = train[,-1]  # drop INDEX column
test = read.csv("wine-evaluation-data.csv")

#DATA EXPLORATION CODE
# make a display df for the project
variable_names = colnames(train)
display_df = data.frame(variable_names)
kable(display_df)

# summarize predictors and check NA
summary(train[,-1])
check_NA(train)

# summarize TARGET
summary(train[1])
make_density_plots(train[1])

# check correlations
make_cor_heatmap(train)

# DATA PREP CODE

# make STARS NA dummy var
train$STARS_dummyNA = ifelse(is.na(train$STARS),1,0)
train$STARS[is.na(train$STARS)] = median(train$STARS, na.rm = T)

# Code to impute data -- DO NOT EVALUATE IN RMD
imputed_train = mice(train, m=1,maxit=10,meth='pmm',seed=500)
```

```r
densityplot(imputed_train) # imputations look good...
train2 = complete(imputed_train,1)
#save(train2,file = 'train2.rds')      <- save load time on knit

#setwd("~/Documents/CUNY/Data_Mining_621/Homework5")  <- this needed here or knit breaks
#load('train2.rds')

# Split into two datasets for zero inflated subset model and set target class in main dataset
train_0 <- train2[train2$TARGET==0,]
train_rest <- train2[train2$TARGET!=0,]

train2$TARGET_CLASS <- ifelse(train2$TARGET==0, 0, 1)

make_cor_heatmap(train2)  # final eval with heatmap

# MODEL building

# Poisson 1
pois_all = glm(TARGET~FixedAcidity + VolatileAcidity + CitricAcid +ResidualSugar+
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
    Density+pH+Sulphates+Alcohol+LabelAppeal +
    AcidIndex +  STARS * STARS_dummyNA , family=poisson,data=train2)

#stepAIC(pois_all)

# results of stepAIC()
pois1 = glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS + STARS_dummyNA, family = poisson,
    data = train2)



summary(pois1)  # INTERPRET RESULTS

## TWO PART ZERO INFLATED MODELS
# BINARY Fit on Target Class
binary_all <- glm(TARGET_CLASS~FixedAcidity + VolatileAcidity +CitricAcid +
                        ResidualSugar+Chlorides +FreeSulfurDioxide +TotalSulfurDioxide  +
                        Density+pH+Sulphates+Alcohol+LabelAppeal+AcidIndex+ STARS * STARS_dummyNA,
                    family="binomial",data=train2)
# stup AIC on model
#stepAIC(binary_all)

binary1 = glm(formula = TARGET_CLASS ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, family = "binomial", data = train2)

summary(binary1)

## TWO PART ZERO INFLATED MODELS
# POISSON using stepAIC
zero_pois_all = glm(TARGET~FixedAcidity + VolatileAcidity +CitricAcid +
```

```
                        ResidualSugar+Chlorides +FreeSulfurDioxide +TotalSulfurDioxide  +
                        Density+pH+Sulphates+Alcohol+LabelAppeal+AcidIndex+ STARS * STARS_dummyNA,
                    family=poisson,data=train_rest)
#stepAIC(zero_pois_all)

zero_pois1 = glm(formula = TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, family = poisson, data = train_rest)

summary(zero_pois1)


# USING zeroinfl function
zero_pois2 = zeroinfl( TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA |
   VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA
        ,data=train2, link="logit", dist="poisson")


display_df = data.frame(summary(zero_pois2)$coefficients$zero)
names(display_df) = c("Coefficient","Std_Error","Z_Score","P_value")

summary(zero_pois1)

### Negative Binomial Models

# initial model... all variablesSTARS_dummyNA as associated with STARS
nb_all = glm.nb(TARGET~FixedAcidity + VolatileAcidity + CitricAcid +ResidualSugar+
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +Density+pH+Sulphates+Alcohol+LabelAppeal +
    AcidIndex +  STARS * STARS_dummyNA,data=train2)

#stepAIC(nb_all)

# results of stepAIC()
nb1 = glm.nb(formula = TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
    ResidualSugar + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
    Density + pH + Sulphates + Alcohol + LabelAppeal + AcidIndex +
    STARS + STARS_dummyNA, data = train2, init.theta = 40606.39252,
    link = log)

summary(nb1)

nbzero_all = glm.nb(TARGET ~FixedAcidity + VolatileAcidity + CitricAcid +ResidualSugar+
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density+pH+Sulphates+Alcohol+LabelAppeal +
    AcidIndex +  STARS * STARS_dummyNA, data=train_rest)


#Sekhar: I obtained the following model:
zero_nb1 = glm.nb(formula = TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, data = train_rest, init.theta = 347549.2109,
    link = log)
```

```
summary(zero_nb1)


zero_nb2 = zeroinfl( TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA |
       VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA
,data=train2, link="logit", dist="negbin")


display_df = data.frame(summary(zero_nb2)$coefficients$zero)
names(display_df) = c("Coefficient","Std_Error","Z_Score","P_value")
kable(display_df)

## Multivariate Linear Regression
#model 1
lm_all = lm(TARGET~FixedAcidity + VolatileAcidity + CitricAcid +ResidualSugar+
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +Density+pH+Sulphates+Alcohol+LabelAppeal +
    AcidIndex +  STARS * STARS_dummyNA, data=train2)

#stepAIC(lm_all)

glm_fit1 = glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, data = train2)

#model 2
err <- vector()

for(i in 1:3)
  {
glm_fit2 <- glm(formula = TARGET ~ poly(VolatileAcidity,i) + poly(Chlorides,i) + poly(FreeSulfurDioxide
    poly(TotalSulfurDioxide,i) + poly(Density,i) + poly(pH,i) + poly(Alcohol,i) + poly(LabelAppeal,i) +
    poly(AcidIndex,i) + poly(STARS,i) + STARS_dummyNA, data = train2)

err[i] <- cv.glm(train2,glm_fit2,K=5)$delta[1]
}

plot(err)



## Model Selection code
Pois_1_err <- vector()
Pois_2_err <- vector()
Pois_3_err <- vector()
lin_1_err <- vector()
for(i in 1:10)
  {
obs = sample(1000)

test_temp = train2[obs,]
```

```
#Developing _poisson-model-1_
pois1 = glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS + STARS_dummyNA, family = poisson,
    data = train2[-obs,])

pred = round(predict(pois1,newdata=test_temp,type="response"))

Pois_1_err[i] <- sqrt(mean((test_temp$TARGET - pred)^2))

x = train2[-obs,]
binary1 = glm(formula = TARGET_CLASS ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, family = "binomial", data = train2[-obs,])

zero_pois1 = glm(formula = TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, family = poisson, data = train2[-obs,][x$TARGET > 0,])


prob = predict(binary1,newdata=test_temp,type="response")
class = ifelse(prob>=0.5,1,0)

pred = round(class * predict(zero_pois1,newdata=test_temp,type="response"))

Pois_2_err[i] <- sqrt(mean((test_temp$TARGET - pred)^2))




zero_pois2 = zeroinfl( TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA |
   VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + pH + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA
        ,data=train2[-obs,], link="logit", dist="poisson")

pred = predict(zero_pois2,newdata=test_temp,type="response")

Pois_3_err[i] <- sqrt(mean((test_temp$TARGET - pred)^2))


glm_fit1 = glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Alcohol + LabelAppeal +
    AcidIndex + STARS + STARS_dummyNA, data = train2[-obs,])

pred = round(predict(glm_fit1,newdata=test_temp,type="response"))

lin_1_err[i] <- sqrt(mean((test_temp$TARGET - pred)^2))

}

display_df <- data.frame(c("Plain Poisson",
                           "Hurdle Poisson", "Zero infl Poisson", "Linear model"),
```

```
        Error = c(mean(Pois_1_err),mean(Pois_2_err), mean(Pois_3_err),mean(lin_1_err)))
```