HMW 5- Data 621

OMER OZEREN

Table of Contents

## Summary

The goal of the homework is to test approaches for evaluating the data set containing various wine characteristics and to predict the number of cases ordered. This project includes comparison of linear, poisson, negative binomial regression models (including zero-inflated negative binomial model).

## Data Exploration

The data set includes 12,795 observations with 14 variables (excluding the target variable).

#### Summary of Variables

Dictionary for Wine Data varianles:

* AcidIndex: Proprietary method of testing
* Alcohol: Alcohol content of wine.
* Chlorides: Chloride content of wine.
* CitricAcid: Citric acid content of wine.
* Density: Density of wine.
* FixedAcidity: Fixed Acidity of wine.
* FreeSulfurDioxide: Sulfur dioxide content of wine.
* LabelAppeal: Marketing score indicating the appeal of label design for consumers.
* ResidualSugar: Residual sugar of wine.
* STARS: Wine rating by a team of experts. Ranges from 1 (Poor) to 4 (Excellent) stars.
* Sulphates: Sulfate content of wine.
* TotalSulfurDioxide: Total sulfur dDioxide of wine.
* VolatileAcidity: Volatile acid content of wine.
* pH: pH of wine

The SUMMARY of Variables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable | Class | Min | Median | Mean | SD | Max |
| FixedAcidity | numeric | -18.1 | 6.9 | 7.076 | 6.318 | 34.4 |
| VolatileAcidity | numeric | -2.79 | 0.28 | 0.3241 | 0.784 | 3.68 |
| CitricAcid | numeric | -3.24 | 0.31 | 0.3084 | 0.8621 | 3.86 |
| ResidualSugar | numeric | -127.8 | 3.9 | 5.419 | 33.75 | 141.2 |
| Chlorides | numeric | -1.171 | 0.046 | 0.05482 | 0.3185 | 1.351 |
| FreeSulfurDioxide | numeric | -555 | 30 | 30.85 | 148.7 | 623 |
| TotalSulfurDioxide | numeric | -823 | 123 | 120.7 | 231.9 | 1057 |
| Density | numeric | 0.8881 | 0.9945 | 0.9942 | 0.02654 | 1.099 |
| pH | numeric | 0.48 | 3.2 | 3.208 | 0.6797 | 6.13 |
| Sulphates | numeric | -3.13 | 0.5 | 0.5271 | 0.9321 | 4.24 |
| Alcohol | numeric | -4.7 | 10.4 | 10.49 | 3.728 | 26.5 |
| LabelAppeal | integer | -2 | 0 | -0.009066 | 0.8911 | 2 |
| AcidIndex | integer | 4 | 8 | 7.773 | 1.324 | 17 |
| STARS | integer | 1 | 2 | 2.042 | 0.9025 | 4 |

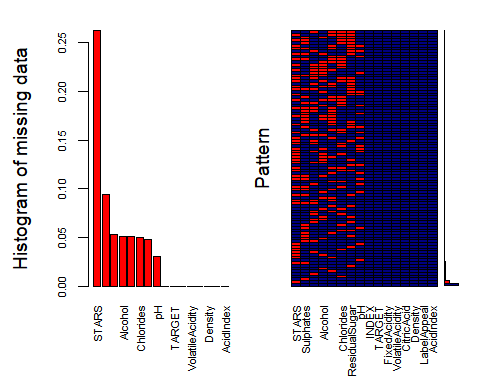
|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Num of NAs | Num of Zeros | Num of Neg Values |
| FixedAcidity | 0 | 39 | 1621 |
| VolatileAcidity | 0 | 18 | 2827 |
| CitricAcid | 0 | 115 | 2966 |
| ResidualSugar | 616 | 6 | 3136 |
| Chlorides | 638 | 5 | 3197 |
| FreeSulfurDioxide | 647 | 11 | 3036 |
| TotalSulfurDioxide | 682 | 7 | 2504 |
| Density | 0 | 0 | 0 |
| pH | 395 | 0 | 0 |
| Sulphates | 1210 | 22 | 2361 |
| Alcohol | 653 | 2 | 118 |
| LabelAppeal | 0 | 5617 | 3640 |
| AcidIndex | 0 | 0 | 0 |
| STARS | 3359 | 0 | 0 |

Chacteristics of Variables:

All but three independent variables are continous. Variables LabelAppeal, AcidIndex and STARS are categorical, but represented by numeric values in logical order.

#### Missing Values

Majority variables have negative values. Eight variables have some NA values. The plot and table below show how the missing values are spread out within the data set. About quarter of observations are missing a STARS value. The rest of variables contain missing values for at most 9.5% of observations.

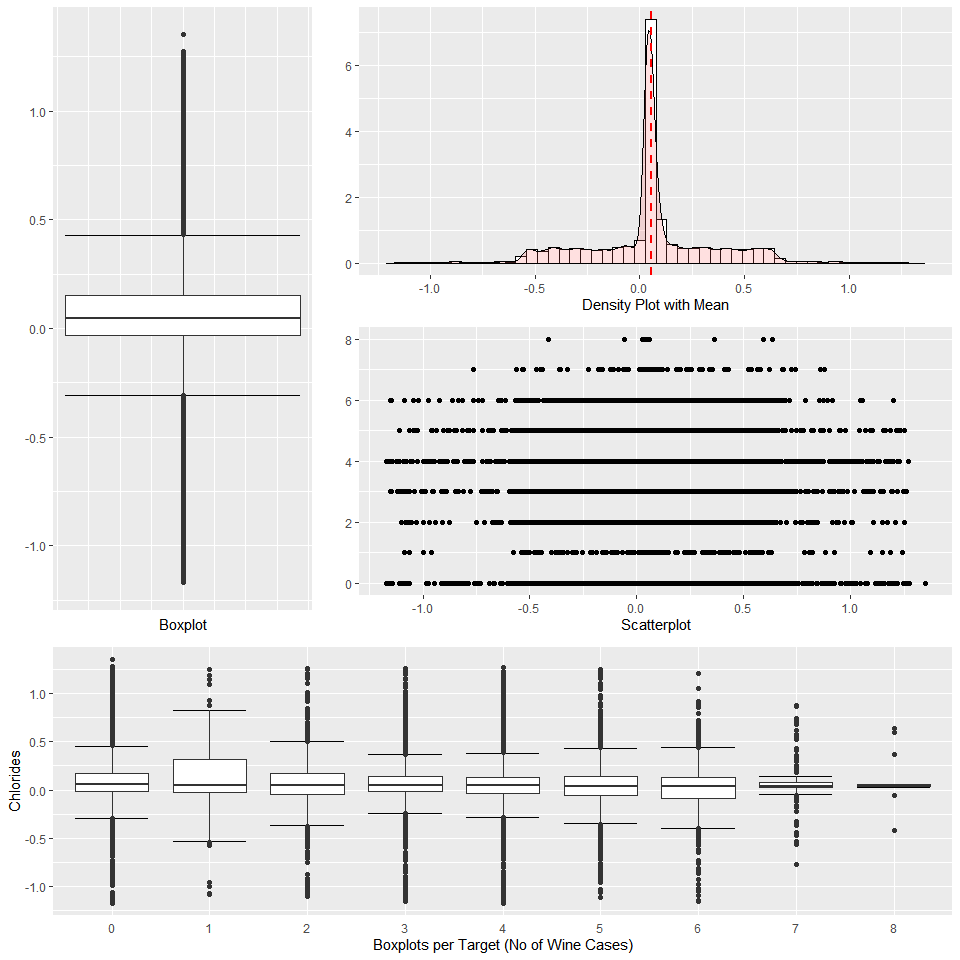


##   
## Variables sorted by number of missings:   
## Variable Count  
## STARS 0.26252442  
## Sulphates 0.09456819  
## TotalSulfurDioxide 0.05330207  
## Alcohol 0.05103556  
## FreeSulfurDioxide 0.05056663  
## Chlorides 0.04986323  
## ResidualSugar 0.04814381  
## pH 0.03087143  
## INDEX 0.00000000  
## TARGET 0.00000000  
## FixedAcidity 0.00000000  
## VolatileAcidity 0.00000000  
## CitricAcid 0.00000000  
## Density 0.00000000  
## LabelAppeal 0.00000000  
## AcidIndex 0.00000000

#### Plots

All dependent variables were inspected using boxplots, density plots and scatterplots. Distribution is similar for all variables - unimodal and symmetrical. Boxplots are also very similar across all possible outcomes with the exception of the last category - 8 cases purchased.

Plots below illustrate results for Chlorides.



#### Correlation Matrix

Correlation matrix below shows that there is very little correlation between variables.This is a good indicator for no-multicollineraity problem.

#### Dependent Variable

The dependent variable TARGET ranges from 0 (no cases purchased) to 8 cases of wine purchased. The most common outcome is 4 cases at 25% of all observations followed closely with no purchase (0 cases) at 21%. Not counting the 0 outcome, it seems that the variable has unimodal, symmetrical distribution resembling normal distibution centered around 4.

|  |  |  |
| --- | --- | --- |
| Outcome | No of Observations | Percent of Total |
| 0 | 2734 | 0.21 |
| 1 | 244 | 0.02 |
| 2 | 1091 | 0.09 |
| 3 | 2611 | 0.2 |
| 4 | 3177 | 0.25 |
| 5 | 2014 | 0.16 |
| 6 | 765 | 0.06 |
| 7 | 142 | 0.01 |
| 8 | 17 | 0 |

## Data Preparation

Two main areas to consider are missing and negative values.

#### Missing Values

The STARS variables contains 3,359 missing values. It represents ratings. **For this analysis missing values for STARS have been replaced with 0.**

Two other categorical variables - LabelAppeal and AcidIndex - do not have any missing values.

Other variables with missing values are good candidates for imputation. Imputation has been done using the mice R package and its method norm.

#### Negative Values

Alcohol variable has a few negative values - only 118 observations. Negative values for this variable are not possible since 0 would be a non-alcoholic beverage. **The variable has been transformed by taking absolute value of all observations.**

For other variables there is significantly more observations with negative values.

#### Training/Testing Split

Data set has been split into a training (75% of observations) and testing (25% of observations) sets. Splitting has been accomplished using the caTools R package based on the TARGET variable to make sure that each set has a proportional number of various target classes.

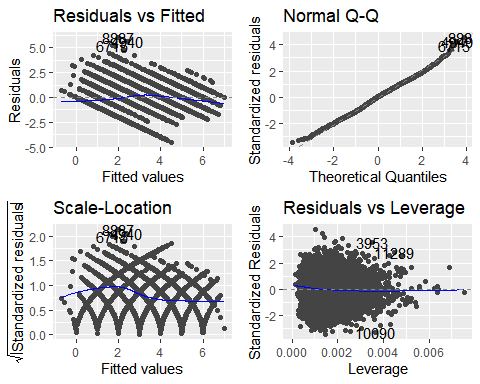
## Modeling: Linear

Two main linear models developed were developed and analyzed. The first one included all available variables. It resulted in R^2 of 0.5268, RMSE of 1.3184 and accuracy in predicting the outcomes in the testing set of 0.2853. The second model used stepwise process in both directions to optimize the model (using the stepAIC function). It resulted in R^2 of 0.5266, RMSE of 1.3193 and accuracy of 0.2847. It appears that the full model performed very slightly better than the stepwise model.

Below is the summary of the full model.

##   
## Call:  
## lm(formula = TARGET ~ . - INDEX, data = wineTRAIN)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.5069 -0.9507 0.0674 0.9089 6.0046   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.674e+00 5.200e-01 7.065 1.72e-12 \*\*\*  
## FixedAcidity 1.736e-04 2.189e-03 0.079 0.936782   
## VolatileAcidity -9.128e-02 1.732e-02 -5.270 1.39e-07 \*\*\*  
## CitricAcid 2.278e-02 1.576e-02 1.445 0.148377   
## ResidualSugar 1.936e-04 4.040e-04 0.479 0.631779   
## Chlorides -1.148e-01 4.279e-02 -2.682 0.007338 \*\*   
## FreeSulfurDioxide 3.203e-04 9.067e-05 3.532 0.000414 \*\*\*  
## TotalSulfurDioxide 1.077e-04 5.848e-05 1.842 0.065486 .   
## Density -6.009e-01 5.121e-01 -1.173 0.240629   
## pH -2.453e-02 1.994e-02 -1.230 0.218579   
## Sulphates -3.981e-02 1.456e-02 -2.735 0.006252 \*\*   
## Alcohol 1.340e-02 3.763e-03 3.562 0.000370 \*\*\*  
## LabelAppeal 4.193e-01 1.575e-02 26.618 < 2e-16 \*\*\*  
## AcidIndex -2.016e-01 1.068e-02 -18.876 < 2e-16 \*\*\*  
## STARS 9.832e-01 1.200e-02 81.959 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.326 on 9580 degrees of freedom  
## Multiple R-squared: 0.5268, Adjusted R-squared: 0.5261   
## F-statistic: 761.7 on 14 and 9580 DF, p-value: < 2.2e-16

Looking at the diagnostic plots we can see that the model performs reasonably well.



The accuracy is fairly low at 28.53%; however, if we examine full confusion matrix below we can see that the model mostly errors only by 1 or 2 cases which may be reasonable enough for a business application.

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1 2 3 4 5 6 7 8  
## 0 24 2 1 3 0 0 0 0 0  
## 1 313 29 68 80 22 7 0 0 0  
## 2 256 26 115 178 109 28 10 2 0  
## 3 86 3 74 236 223 71 9 0 0  
## 4 5 1 14 129 301 197 42 3 0  
## 5 0 0 1 27 128 155 76 14 1  
## 6 0 0 0 0 11 45 51 15 1  
## 7 0 0 0 0 0 1 3 2 2  
## 8 0 0 0 0 0 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.2853   
## 95% CI : (0.2697, 0.3013)  
## No Information Rate : 0.2481   
## P-Value [Acc > NIR] : 8.841e-07   
##   
## Kappa : 0.1642   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5  
## Sensitivity 0.035088 0.475410 0.42125 0.36141 0.37909 0.30754  
## Specificity 0.997615 0.843899 0.79194 0.81704 0.83749 0.90838  
## Pos Pred Value 0.800000 0.055877 0.15884 0.33618 0.43497 0.38557  
## Neg Pred Value 0.791798 0.988064 0.93619 0.83307 0.80343 0.87527  
## Prevalence 0.213750 0.019062 0.08531 0.20406 0.24813 0.15750  
## Detection Rate 0.007500 0.009062 0.03594 0.07375 0.09406 0.04844  
## Detection Prevalence 0.009375 0.162188 0.22625 0.21937 0.21625 0.12562  
## Balanced Accuracy 0.516351 0.659655 0.60659 0.58922 0.60829 0.60796  
## Class: 6 Class: 7 Class: 8  
## Sensitivity 0.26702 0.055556 0.00000  
## Specificity 0.97607 0.998104 1.00000  
## Pos Pred Value 0.41463 0.250000 NaN  
## Neg Pred Value 0.95450 0.989348 0.99875  
## Prevalence 0.05969 0.011250 0.00125  
## Detection Rate 0.01594 0.000625 0.00000  
## Detection Prevalence 0.03844 0.002500 0.00000  
## Balanced Accuracy 0.62154 0.526830 0.50000

## Modeling: Poisson

The linear model seemed to perform good with all variables. So for the poisson regression similar strategy was applied - a model with all variables and a model optimized by the stepwise method.RMSE for this model is 1.39855, slightly worse than for the linear model.

##   
## Call:  
## glm(formula = TARGET ~ . - INDEX, family = poisson, data = wineTRAIN)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9560 -0.7237 0.0709 0.5750 3.2336   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.416e+00 2.255e-01 6.280 3.40e-10 \*\*\*  
## FixedAcidity -2.847e-04 9.536e-04 -0.299 0.765268   
## VolatileAcidity -3.185e-02 7.538e-03 -4.225 2.39e-05 \*\*\*  
## CitricAcid 8.497e-03 6.825e-03 1.245 0.213161   
## ResidualSugar 1.864e-05 1.756e-04 0.106 0.915441   
## Chlorides -3.928e-02 1.860e-02 -2.111 0.034737 \*   
## FreeSulfurDioxide 1.316e-04 3.924e-05 3.354 0.000797 \*\*\*  
## TotalSulfurDioxide 4.204e-05 2.562e-05 1.641 0.100760   
## Density -2.169e-01 2.213e-01 -0.980 0.327046   
## pH -1.222e-02 8.648e-03 -1.413 0.157722   
## Sulphates -1.571e-02 6.336e-03 -2.479 0.013185 \*   
## Alcohol 3.316e-03 1.642e-03 2.020 0.043396 \*   
## LabelAppeal 1.298e-01 6.997e-03 18.553 < 2e-16 \*\*\*  
## AcidIndex -8.412e-02 5.256e-03 -16.004 < 2e-16 \*\*\*  
## STARS 3.133e-01 5.187e-03 60.395 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 17142 on 9594 degrees of freedom  
## Residual deviance: 11060 on 9580 degrees of freedom  
## AIC: 35045  
##   
## Number of Fisher Scoring iterations: 5

Confusion matrix below comparing predicted values to the test data shows that the model does not predict *no purchase* outcome (count is 0).Accuracy is lower than for the linear model.

## Accuracy   
## 0.233125

## Reference  
## Prediction 0 1 2 3 4 5 6 7 8 9 10  
## 0 0 0 0 0 0 0 0 0 0 0 0  
## 1 204 19 35 39 6 4 0 0 0 0 0  
## 2 421 38 166 282 178 45 12 2 0 0 0  
## 3 58 4 64 231 305 118 12 1 0 0 0  
## 4 1 0 6 75 186 155 36 2 0 0 0  
## 5 0 0 2 24 94 103 53 11 1 0 0  
## 6 0 0 0 2 18 53 32 6 0 0 0  
## 7 0 0 0 0 5 19 28 9 1 0 0  
## 8 0 0 0 0 2 6 17 4 0 0 0  
## 9 0 0 0 0 0 1 0 1 2 0 0  
## 10 0 0 0 0 0 0 1 0 0 0 0

## Modeling: Negative Binomial

Using the MASS R package, negative binomial model was created using all variables. This model turned out to be nearly identical to the poisson model.

## Modeling: Zero-Inflated Negative Binomial

Poisson and negative binomial models do not account for the 0 outcome. So a zero-inflated negative binomial model was attempted using the pscl R package. RMSE for this model is 1.2727, the best one out of all models.

##   
## Call:  
## zeroinfl(formula = TARGET ~ . - INDEX, data = wineTRAIN, dist = "negbin")  
##   
## Pearson residuals:  
## Min 1Q Median 3Q Max   
## -2.105450 -0.406781 -0.007306 0.370589 5.885163   
##   
## Count model coefficients (negbin with log link):  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.387e+00 2.329e-01 5.957 2.57e-09 \*\*\*  
## FixedAcidity 3.228e-04 9.820e-04 0.329 0.742387   
## VolatileAcidity -1.187e-02 7.783e-03 -1.526 0.127118   
## CitricAcid 2.758e-03 7.006e-03 0.394 0.693862   
## ResidualSugar -1.059e-04 1.812e-04 -0.584 0.558888   
## Chlorides -2.436e-02 1.911e-02 -1.274 0.202495   
## FreeSulfurDioxide 1.724e-05 3.963e-05 0.435 0.663490   
## TotalSulfurDioxide -3.882e-05 2.549e-05 -1.523 0.127695   
## Density -2.332e-01 2.280e-01 -1.023 0.306351   
## pH 3.883e-03 8.915e-03 0.436 0.663155   
## Sulphates -1.098e-03 6.521e-03 -0.168 0.866280   
## Alcohol 7.189e-03 1.679e-03 4.282 1.85e-05 \*\*\*  
## LabelAppeal 2.321e-01 7.276e-03 31.891 < 2e-16 \*\*\*  
## AcidIndex -1.777e-02 5.663e-03 -3.137 0.001707 \*\*   
## STARS 1.028e-01 5.964e-03 17.238 < 2e-16 \*\*\*  
## Log(theta) 1.694e+01 4.783e+00 3.541 0.000399 \*\*\*  
##   
## Zero-inflation model coefficients (binomial with logit link):  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.4586261 1.5648921 -2.849 0.004383 \*\*   
## FixedAcidity -0.0002330 0.0064138 -0.036 0.971025   
## VolatileAcidity 0.1521666 0.0500158 3.042 0.002347 \*\*   
## CitricAcid -0.0168785 0.0463039 -0.365 0.715474   
## ResidualSugar -0.0014052 0.0011789 -1.192 0.233247   
## Chlorides -0.0439761 0.1239205 -0.355 0.722685   
## FreeSulfurDioxide -0.0009264 0.0002715 -3.412 0.000644 \*\*\*  
## TotalSulfurDioxide -0.0007884 0.0001692 -4.659 3.18e-06 \*\*\*  
## Density 0.6548048 1.5366155 0.426 0.670010   
## pH 0.1873836 0.0580400 3.229 0.001244 \*\*   
## Sulphates 0.1280396 0.0425675 3.008 0.002630 \*\*   
## Alcohol 0.0233215 0.0110156 2.117 0.034249 \*   
## LabelAppeal 0.7547617 0.0495577 15.230 < 2e-16 \*\*\*  
## AcidIndex 0.4345239 0.0300666 14.452 < 2e-16 \*\*\*  
## STARS -2.3833651 0.0698008 -34.145 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1   
##   
## Theta = 22639264.4045   
## Number of iterations in BFGS optimization: 36   
## Log-likelihood: -1.53e+04 on 31 Df

This model has the accuracy of 35.78%, again the best one out of all models. It predicts 0 outcomes (not ideally, but perhaps it can be improved with more research).

## Accuracy   
## 0.358125

## Reference  
## Prediction 0 1 2 3 4 5 6 7 8  
## 0 107 1 0 6 4 3 1 1 0  
## 1 323 15 34 76 39 16 5 1 0  
## 2 138 36 125 117 39 11 4 0 0  
## 3 103 9 107 309 246 64 4 0 0  
## 4 13 0 7 140 373 200 33 3 0  
## 5 0 0 0 5 81 164 75 11 0  
## 6 0 0 0 0 12 36 45 12 0  
## 7 0 0 0 0 0 10 20 8 4  
## 8 0 0 0 0 0 0 4 0 0

## Model Comparison

Considering log-likelihood of all models, it is clear that zero-inflated negative binomial model is the best option. More research in that direction will probably be beneficial.

|  |  |  |
| --- | --- | --- |
|  | Log-Likelihood | DF |
| **Linear** | -16316 | 16 |
| **Poisson** | -17507 | 15 |
| **NB** | -17507 | 16 |
| **ZINB** | -15303 | 31 |

Using full models in all methods allows comparison of coefficients. For the most part coefficients are similar in sign and in magnitude. There are a couple of small coefficients that change signs between NB and ZINB models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Linear | Poisson | NB | ZINB (Count) |
| **(Intercept)** | 3.674 | 1.416 | 1.416 | 1.387 |
| **FixedAcidity** | 0.000174 | -0.000285 | -0.000285 | 0.000323 |
| **VolatileAcidity** | -0.09128 | -0.03185 | -0.03185 | -0.01187 |
| **CitricAcid** | 0.02278 | 0.008497 | 0.008497 | 0.002758 |
| **ResidualSugar** | 0.000194 | 1.9e-05 | 1.9e-05 | -0.000106 |
| **Chlorides** | -0.1148 | -0.03928 | -0.03928 | -0.02436 |
| **FreeSulfurDioxide** | 0.00032 | 0.000132 | 0.000132 | 1.7e-05 |
| **TotalSulfurDioxide** | 0.000108 | 4.2e-05 | 4.2e-05 | -3.9e-05 |
| **Density** | -0.6009 | -0.2169 | -0.2169 | -0.2332 |
| **pH** | -0.02453 | -0.01222 | -0.01222 | 0.003883 |
| **Sulphates** | -0.03981 | -0.01571 | -0.01571 | -0.001098 |
| **Alcohol** | 0.0134 | 0.003316 | 0.003316 | 0.007189 |
| **LabelAppeal** | 0.4193 | 0.1298 | 0.1298 | 0.2321 |
| **AcidIndex** | -0.2016 | -0.08412 | -0.08412 | -0.01777 |
| **STARS** | 0.9832 | 0.3133 | 0.3133 | 0.1028 |

## APPENDIX A: Evaluation Data Set

Please note that this appendix includes first 100 observations from the evaluation set.

|  |  |  |
| --- | --- | --- |
| Index | Predicted Value | Predicted Outcome |
| 3 | 1.894 | 2 |
| 9 | 3.826 | 4 |
| 10 | 2.513 | 3 |
| 18 | 2.492 | 2 |
| 21 | 0.7127 | 1 |
| 30 | 5.695 | 6 |
| 31 | 3.595 | 4 |
| 37 | 1.343 | 1 |
| 39 | 0.2309 | 0 |
| 47 | 1.467 | 1 |
| 60 | 2.693 | 3 |
| 62 | 0.2158 | 0 |
| 63 | 3.539 | 4 |
| 64 | 1.329 | 1 |
| 68 | 1.176 | 1 |
| 75 | 2.761 | 3 |
| 76 | 2.517 | 3 |
| 83 | 0.0539 | 0 |
| 87 | 3.674 | 4 |
| 92 | 5.435 | 5 |
| 98 | 2.317 | 2 |
| 106 | 1.682 | 2 |
| 107 | 0.473 | 0 |
| 113 | 2.537 | 3 |
| 120 | 3.448 | 3 |
| 123 | 5.88 | 6 |
| 125 | 2.838 | 3 |
| 126 | 5.872 | 6 |
| 128 | 4.508 | 5 |
| 129 | 2.449 | 2 |
| 131 | 4.213 | 4 |
| 135 | 0.9485 | 1 |
| 141 | 4.205 | 4 |
| 147 | 3.249 | 3 |
| 148 | 1.312 | 1 |
| 151 | 3.745 | 4 |
| 156 | 3.179 | 3 |
| 157 | 3.38 | 3 |
| 174 | 1.655 | 2 |
| 186 | 0.506 | 1 |
| 193 | 2.592 | 3 |
| 195 | 0.9228 | 1 |
| 212 | 0.6653 | 1 |
| 213 | 0.6854 | 1 |
| 217 | 2.986 | 3 |
| 223 | 3.834 | 4 |
| 226 | 3.157 | 3 |
| 228 | 4.544 | 5 |
| 230 | 4.127 | 4 |
| 241 | 2.628 | 3 |
| 243 | 3.746 | 4 |
| 249 | 1.071 | 1 |
| 281 | 4.079 | 4 |
| 288 | 0.277 | 0 |
| 294 | 1.913 | 2 |
| 295 | 1.963 | 2 |
| 300 | 5.567 | 6 |
| 302 | 4.323 | 4 |
| 303 | 1.856 | 2 |
| 308 | 1.706 | 2 |
| 319 | 4.84 | 5 |
| 320 | 0.9055 | 1 |
| 324 | 2.964 | 3 |
| 331 | 2.666 | 3 |
| 343 | 2.961 | 3 |
| 347 | 2.422 | 2 |
| 348 | 3.685 | 4 |
| 350 | 4.667 | 5 |
| 357 | 1.443 | 1 |
| 358 | 3.605 | 4 |
| 360 | 4.18 | 4 |
| 366 | 3.558 | 4 |
| 367 | 2.621 | 3 |
| 368 | 4.894 | 5 |
| 376 | 2.318 | 2 |
| 380 | 3.275 | 3 |
| 388 | 0.5947 | 1 |
| 396 | 4.574 | 5 |
| 398 | 4.527 | 5 |
| 403 | 3.921 | 4 |
| 410 | 2.043 | 2 |
| 412 | 0.6185 | 1 |
| 420 | 2.501 | 3 |
| 434 | 2.495 | 2 |
| 440 | 3.049 | 3 |
| 450 | 3.489 | 3 |
| 453 | 2.763 | 3 |
| 464 | 4.602 | 5 |
| 465 | 4.51 | 5 |
| 466 | 4.76 | 5 |
| 473 | 2.475 | 2 |
| 476 | 1.708 | 2 |
| 478 | 1.517 | 2 |
| 479 | 3.189 | 3 |
| 493 | 2.733 | 3 |
| 497 | 3.061 | 3 |
| 503 | 3.663 | 4 |
| 504 | 3.592 | 4 |
| 505 | 2.192 | 2 |
| 507 | 0.2199 | 0 |

## APPENDIX B: R Script

# Required libraries  
library(ggplot2) # plotting  
library(dplyr) # data manipulation  
library(gridExtra) # display  
library(knitr) # display  
library(kableExtra) # display  
library(mice) # imputation  
library(caTools) # train-test split  
library(MASS) # boxcox  
library(Metrics) # rmse  
library(caret) # confusion matrix  
library(VIM) # plotting NAs  
library(ggfortify) # plotting lm diagnostic  
library(car) # VIF  
library(pscl) # zero-inflated model  
# Import data  
wine <- read.csv(url(paste0("https://raw.githubusercontent.com/omerozeren/DATA621/master/wine-training-data.csv")),  
 na.strings=c("","NA"))  
colnames(wine)[1] <- "INDEX"  
# Basic statistic  
nrow(wine); ncol(wine)  
summary(wine)  
# Summary table  
sumtbl = data.frame(Variable = character(),  
 Class = character(),  
 Min = integer(),  
 Median = integer(),  
 Mean = double(),  
 SD = double(),  
 Max = integer(),  
 Num\_NAs = integer(),  
 Num\_Zeros = integer(),  
 Num\_Neg = integer())  
for (i in c(3:16)) {  
 sumtbl <- rbind(sumtbl, data.frame(Variable = colnames(wine)[i],  
 Class = class(wine[,i]),  
 Min = min(wine[,i], na.rm=TRUE),  
 Median = median(wine[,i], na.rm=TRUE),  
 Mean = mean(wine[,i], na.rm=TRUE),  
 SD = sd(wine[,i], na.rm=TRUE),  
 Max = max(wine[,i], na.rm=TRUE),  
 Num\_NAs = sum(is.na(wine[,i])),  
 Num\_Zeros = length(which(wine[,i]==0)),  
 Num\_Neg = sum(wine[,i]<0 & !is.na(wine[,i]))))  
}  
colnames(sumtbl) <- c("Variable", "Class", "Min", "Median", "Mean", "SD", "Max",   
 "Num of NAs", "Num of Zeros", "Num of Neg Values")  
sumtbl  
# Categorical variables  
table(wine$LabelAppeal)  
table(wine$AcidIndex)  
table(wine$STARS)  
# Exploratory plots  
v <- "FixedAcidity"  
v <- "VolatileAcidity"  
v <- "CitricAcid"  
v <- "ResidualSugar"  
v <- "Chlorides"  
v <- "FreeSulfurDioxide"  
v <- "TotalSulfurDioxide"  
v <- "Density"  
v <- "pH"  
v <- "Sulphates"  
v <- "Alcohol"  
v <- "LabelAppeal"   
v <- "AcidIndex"  
v <- "STARS"  
pd <- as.data.frame(cbind(wine[, v], wine$TARGET)); colnames(pd) <- c("X", "Y")  
bp <- ggplot(pd, aes(x = 1, y = X)) + stat\_boxplot(geom ='errorbar') +   
 geom\_boxplot() +   
 xlab("Boxplot") + ylab("") + theme(axis.text.x=element\_blank(),   
 axis.ticks.x=element\_blank())  
hp <- ggplot(pd, aes(x = X)) + geom\_histogram(aes(y=..density..),   
 colour="black", fill="white") +  
 geom\_density(alpha=.2, fill="#FF6666") + ylab("") +   
 xlab("Density Plot with Mean") +  
 geom\_vline(aes(xintercept=mean(X, na.rm=TRUE)),   
 color="red", linetype="dashed", size=1)  
sp <- ggplot(pd, aes(x=X, y=Y)) + geom\_point() + xlab("Scatterplot")  
grid.arrange(bp, hp, sp, layout\_matrix=rbind(c(1,2,2),c(1,3,3)))  
ggplot(wine, aes(x = as.factor(TARGET), y = Chlorides)) +   
 stat\_boxplot(geom ='errorbar') + geom\_boxplot() +   
 xlab("Boxplots per No of Wine Cases") + ylab("pH") +   
 theme(axis.ticks.x=element\_blank())  
# Correlation matrix  
cm <- cor(wine[,2:16], use="pairwise.complete.obs")  
cm <- round(cm, 2)  
cmout <- as.data.frame(cm) %>% mutate\_all(function(x) {  
 cell\_spec(x, "html", color = ifelse(x>0.5 | x<(-0.5),"blue","black"))  
})  
rownames(cmout) <- colnames(cmout)  
cmout %>%  
 kable("html", escape = F, align = "c", row.names = TRUE) %>%  
 kable\_styling("striped", full\_width = F)  
# IMPUTATION / TRANSFORMATION  
wineOriginal <- wine # Backup of original data  
wine$STARS[is.na(wine$STARS)] <- 0 # Missing STARS are 0 score  
# Missing values - table  
md.pattern(wine)  
# Missing values - plot  
aggr\_plot <- aggr(wine, col=c('navyblue','red'),   
 numbers=FALSE, sortVars=TRUE, labels=names(wine),   
 cex.axis=.7, gap=3,   
 ylab=c("Histogram of missing data","Pattern"))  
# Imputation  
wineImputed <- mice(wine, m=5, maxit=20, meth='norm', seed=500)  
summary(wineImputed)  
wine <- complete(wineImputed)  
summary(wine)  
# Proportion of target variable  
table(wine$TARGET)  
table(wine$TARGET)/sum(table(wine$TARGET))  
wine$Alcohol <- abs(wine$Alcohol)  
# Split into train and validation sets  
set.seed(88)  
split <- sample.split(wine$TARGET, SplitRatio = 0.75)  
wineTRAIN <- subset(wine, split == TRUE)  
wineTEST <- subset(wine, split == FALSE)  
table(wineTRAIN$TARGET)/sum(table(wineTRAIN$TARGET))  
# LINEAR MODEL  
# All variables  
lmModel <- lm(TARGET ~ .-INDEX,data = wineTRAIN)  
summary(lmModel)  
# stepAIC  
lmModel <- stepAIC(lmModel, trace=FALSE, direction='both')  
summary(lmModel)  
# Model returned by step AIC  
lmModel <- lm(TARGET ~ VolatileAcidity + CitricAcid +   
 Chlorides + FreeSulfurDioxide +   
 TotalSulfurDioxide + Sulphates + Alcohol +   
 LabelAppeal + AcidIndex + STARS,  
 data = wineTRAIN)  
summary(lmModel)  
# Manual variations  
lmModel <- lm(TARGET ~ VolatileAcidity + Chlorides +   
 FreeSulfurDioxide +   
 TotalSulfurDioxide + Sulphates + Alcohol +   
 LabelAppeal + AcidIndex + STARS,  
 data = wineTRAIN)  
summary(lmModel)  
lmModel <- lm(TARGET ~ VolatileAcidity + Chlorides +   
 FreeSulfurDioxide +   
 TotalSulfurDioxide + Alcohol +   
 LabelAppeal + AcidIndex + STARS,  
 data = wineTRAIN)  
summary(lmModel)  
# Calculate RMSE  
pred <- predict(lmModel, newdata=wineTEST)  
rmse(wineTEST$TARGET, pred)  
# Confusion matrix  
predRound <- as.factor(round(pred,0))  
table(predRound)  
levels(predRound) <- levels(as.factor(wineTEST$TARGET))  
confusionMatrix(predRound, as.factor(wineTEST$TARGET))  
autoplot(lmModel)  
# Model plots  
plot(lmModel$residuals, ylab="Residuals")  
abline(h=0)  
plot(lmModel$fitted.values, lmModel$residuals,   
 xlab="Fitted Values", ylab="Residuals")  
abline(h=0)  
qqnorm(lmModel$residuals)  
qqline(lmModel$residuals)  
# POISSON and NB REGRESSION MODEL  
# Poisson 1  
glmModel <- glm (TARGET ~ .-INDEX, data = wineTRAIN, family = poisson)  
summary(glmModel)  
pred <- predict(glmModel, newdata=wineTEST, type='response')  
rmse(wineTEST$TARGET, pred)  
predRound <- as.factor(round(pred,0))  
testData <- as.factor(wineTEST$TARGET)  
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")  
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")  
confusionMatrix(predRound, testData)  
# Poisson 2  
glmModel2 <- stepAIC(glmModel, trace=FALSE, direction='both')  
summary(glmModel2)  
pred <- predict(glmModel2, newdata=wineTEST, type='response')  
rmse(wineTEST$TARGET, pred)  
predRound <- as.factor(round(pred,0))  
testData <- as.factor(wineTEST$TARGET)  
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")  
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")  
confusionMatrix(predRound, testData)  
# Poisson 3  
glmModel3 <- glm(TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +   
 Sulphates + Alcohol + LabelAppeal +   
 AcidIndex + STARS, family = poisson, data = wineTRAIN)  
summary(glmModel3)  
pred <- predict(glmModel3, newdata=wineTEST, type='response')  
rmse(wineTEST$TARGET, pred)  
predRound <- as.factor(round(pred,0))  
testData <- as.factor(wineTEST$TARGET)  
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")  
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")  
confusionMatrix(predRound, testData)  
# NB  
nbModel <- glm.nb(TARGET ~ .-INDEX, data = wineTRAIN)  
summary(nbModel)  
pred <- predict(nbModel, newdata=wineTEST, type='response')  
rmse(wineTEST$TARGET, pred)  
predRound <- as.factor(round(pred,0))  
testData <- as.factor(wineTEST$TARGET)  
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")  
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")  
confusionMatrix(predRound, testData)  
# Zero Inflated  
zrModel <- zeroinfl(TARGET ~ .-INDEX, data = wineTRAIN, dist = "negbin")  
summary(zrModel)  
pred <- predict(zrModel, newdata=wineTEST, type='response')  
rmse(wineTEST$TARGET, pred)  
predRound <- as.factor(round(pred,0))  
testData <- as.factor(wineTEST$TARGET)  
confusionMatrix(predRound, testData)  
# Deviance residuals  
anova(glmModel, test="Chisq")  
anova(glmModel2, test="Chisq")  
anova(glmModel3, test="Chisq")  
anova(nbModel, test="Chisq")  
anova(zrModel, test="Chisq")  
# VIF  
vif(glmModel)  
vif(nbModel)  
vif(zrModel)  
# Coefficients  
coef <- as.data.frame(lmModel$coefficients)  
coef <- cbind(coef, as.data.frame(glmModel$coefficients))  
coef <- cbind(coef, as.data.frame(nbModel$coefficients))  
coef <- cbind(coef, as.data.frame(zrModel$coefficients))  
# Prediction  
eval <- read.csv(url(paste0("https://raw.githubusercontent.com/omerozeren/DATA621/master/wine-evaluation-data.csv")),  
 na.strings=c("","NA"))  
colnames(eval)[1] <- "INDEX"  
sumtbl = data.frame(Variable = character(),  
 Class = character(),  
 Min = integer(),  
 Median = integer(),  
 Mean = double(),  
 SD = double(),  
 Max = integer(),  
 Num\_NAs = integer(),  
 Num\_Zeros = integer(),  
 Num\_Neg = integer())  
for (i in c(3:16)) {  
 sumtbl <- rbind(sumtbl, data.frame(Variable = colnames(eval)[i],  
 Class = class(eval[,i]),  
 Min = min(eval[,i], na.rm=TRUE),  
 Median = median(eval[,i], na.rm=TRUE),  
 Mean = mean(eval[,i], na.rm=TRUE),  
 SD = sd(eval[,i], na.rm=TRUE),  
 Max = max(eval[,i], na.rm=TRUE),  
 Num\_NAs = sum(is.na(eval[,i])),  
 Num\_Zeros = length(which(eval[,i]==0)),  
 Num\_Neg = sum(eval[,i]<0 & !is.na(eval[,i]))))  
}  
colnames(sumtbl) <- c("Variable", "Class", "Min", "Median", "Mean", "SD", "Max",   
 "Num of NAs", "Num of Zeros", "Num of Neg Values")  
sumtbl  
eval$STARS[is.na(eval$STARS)] <- 0  
eval$Alcohol <- abs(eval$Alcohol)  
evalImputed <- mice(eval, m=5, maxit=10, meth='norm', seed=500)  
eval <- complete(evalImputed)  
pred <- predict(zrModel, newdata=eval, type="response")  
results <- eval[, c("INDEX")]  
results <- cbind(results, prob=round(pred,4))  
results <- cbind(results, predict=round(pred,0))  
colnames(results) <- c("Index", "Predicted Value", "Predicted Outcome")  
pander(head(results, 100))