The Prediction of Red Wine Quality

Gizem Altun

3/25/23

Supervised Learning: Linear Regression Models

In this task, the following steps were followed and the regression was dealt with.

1. Definition of the Problem, Target and Features
2. Describing the Data Set
3. Splitting the Data Set(Train and Test Set)
4. Training a Linear Regression Model
5. Measuring Model Performance
6. Checking the Possible Over and Underfitting Problem
7. Predicting the Target Variable with the New Observation

1. Definition of the Problem, Target and Features

In this task, two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal.Using this dataset, we can predict the red wine quality.

Target variable and features were examined in this dataset. A dependent variable(target) is a variable that is observed to change in response to independent variables(features). Then the target variable in this data set is the "quality" variable.(Y: wine quality) Since we have a target variable in this dataset, this process as "supervised learning" process. Quality takes values between three and eight. For this reason, target variable has been numeric and takes any values in range so continuous in this task.Since the target variable takes continuous and numeric values, this has been considered as a "regression problem". To find the model shows us relationship between the target variable and the features we can use "linear regression models".

In addition, aside from using regression modelling, is to set an arbitrary cutoff for dependent variable (wine quality) at e.g. 7 or higher getting classified as 'good/1' and the remainder as ‘not good/0’. In this way, the target variable converted into a categorical variable can be predicted using a logistic regression model. However, this problem has been considered as a linear regression model.

Independent variables(features) are variables that cause a change in dependent variable(target). The features in this dataset are; X1:fixed acidity, X2:volatile acidity, X3:citric acid, X4:residual sugar, X5:chlorides, X6:free sulfur dioxide, X7:total sulfur dioxide, X8:density, X9:pH, X10:sulphates, X11:alcohol. There are a total of 11 features and 1 target variable(quality) in this task. Since there is more than one X variable(features), this model is called a “Multiple Linear Regression Model”.

Packages

Before proceeding to the steps written above about regression, the packages required for these steps have been installed.As a first step, the packages were installed. As a second step, he was called from the library.

```
install.packages("readr") #This package allows us to read our dataset.
install.packages("ggplot2") #This package allows us to visualize our dataset.
install.packages("car") #This package allows to check the assumption of linear regression
library(readr)
library(ggplot2)
library(car)
```

2. Describing the Data Set

This dataset related to red and white vinho verde wine samples, from the north of Portugal. This dataset is taken from Kaggle.(https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009?resource=download) The readr package was used because the data set’s file is of the csv file type. At the same time, import dataset from environment window to add the dataset.

```
redwine <- read_csv("winequality-red.csv")
```

The data set was looked at using the str() function. This function gives us the following information; number of observation, number of features, name of features, type of features, a few observations of features.

```
str(redwine)
```

```
spc_tbl_ [1,599 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ fixed_acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile_acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric_acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual_sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0
 $ free_sulfur_dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
 $ total_sulfur_dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...
 - attr(*, "spec")=
  .. cols(
  ..   fixed_acidity = col_double(),
  ..   volatile_acidity = col_double(),
  ..   citric_acid = col_double(),
  ..   residual_sugar = col_double(),
  ..   chlorides = col_double(),
  ..   free_sulfur_dioxide = col_double(),
  ..   total_sulfur_dioxide = col_double(),
  ..   density = col_double(),
  ..   pH = col_double(),
  ..   sulphates = col_double(),
  ..   alcohol = col_double(),
  ..   quality = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
```

Describing the dataset:

This dataset contains 1599 observation of 12 varibles. This also equates to 1599 rows and 12 columns. Variables:

1. fixed acidity: Numeric and continuous variable.Continuous because, it takes the values between zero and positive infinity it takes any values.
2. volatile acidity: Numeric and continuous variable.
3. citric acid: Numeric and continuous variable.
4. residual sugar: Numeric and continuous variable.
5. chlorides: Numeric and continuous variable.
6. free sulfur dioxide: Numeric and continuous variable.
7. total sulfur dioxide: Numeric and continuous variable.

8. density: Numeric and continuous variable.
9. pH: Numeric and continuous variable.
10. sulphates: Numeric and continuous variable.
11. alcohol: Numeric and continuous variable.
12. quality: Numeric and continuous variable.

From the first variable to the twelfth variable, they are all features. The twelfth variable(quality) is the target variable. As can be seen, the dataset contains the target variable. This process as "supervised learning" process. According to type of the target variable we can conclude that type of task. The type of target variable is continuous and numeric this is called regression task. There is more than one X variable(features), this model is called a "Multiple Linear Regression Model".

```r
dim(redwine)
```

```
[1] 1599   12
```

The dim() function shows us to dimension of dataset. When there are how many observations in the first column, we see how many variables there are in the second column. This dataset contains 1599 observation of 12 varibles.

```r
redwine <- na.exclude(redwine)
```

Before training the model, excluded the all NA variables in the dataset.

3. Splitting the Data Set(Train and Test Set)

Train/ Test split is used rather than just validating the model on train set, it gives also an estimate how well the model performans on new data.That we should train model a dataset then we check performance of model in a different dataset so far doing this. We can seperate our original dataset to train and test set data. We can do this steps by selecting randomly observation so in practise mostly the number of observation in train dataset much more than test data. The ratio like 80% train data, 20% test data. This dataset was also splited at this ratio. The split the dataset is important because it allows us to "generalizability" the model.

```r
set.seed(123)
index <- sample(1 : nrow(redwine), round(nrow(redwine) * 0.80))
train <- redwine[index,]
test <- redwine[-index,]
```

We should split the data set to two different set named train and test set. We will use sample function to create a new object which is index. In this sample we need to use a sequence. When we create an sequence starting with 1 up to the number of rows of the redwine data set. then we use the same dataset to split 80% of the dataset.

4. Training a Linear Regression Model

The process to estimate the model coefficients is called model training. This task predict the target variable.(quality) So, the lm() function was used to train the model.Model formul tips: (y ~ .) Here "y" refers to the target variable, dot refers to the features. Since the dot mark is set, this includes all the features.

```
lrm_model <- lm(quality ~ ., data = train)
lrm_model
```

```
Call:
lm(formula = quality ~ ., data = train)

Coefficients:
         (Intercept)         fixed_acidity      volatile_acidity  
           14.922386              0.012451             -1.021530  
         citric_acid        residual_sugar             chlorides  
           -0.129780              0.006223             -2.058768  
 free_sulfur_dioxide  total_sulfur_dioxide               density  
            0.004125             -0.003556            -10.264153  
                  pH             sulphates               alcohol  
           -0.577992              0.865208              0.290745  
```

This code output gives information about the model. The estimated values of the model parameters which are Betas and the trained data. From this code output, it shows which variables are numerical and which variables are categorical.(name + name of the category) but as can be seen, this entire data set is numeric and continuous. summary() function was used to see more details about the model.

```
summary(lrm_model)
```

```
Call:
```

```
lm(formula = quality ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.65018 -0.38387 -0.04515  0.45703  2.01377 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          1.492e+01  2.405e+01   0.620   0.5351    
fixed_acidity        1.245e-02  2.998e-02   0.415   0.6780    
volatile_acidity    -1.022e+00  1.398e-01  -7.306 4.87e-13 ***
citric_acid         -1.298e-01  1.687e-01  -0.769   0.4419    
residual_sugar       6.223e-03  1.749e-02   0.356   0.7220    
chlorides           -2.059e+00  4.711e-01  -4.370 1.34e-05 ***
free_sulfur_dioxide  4.125e-03  2.501e-03   1.649   0.0994 .  
total_sulfur_dioxide -3.556e-03  8.348e-04  -4.260 2.20e-05 ***
density             -1.026e+01  2.455e+01  -0.418   0.6760    
pH                  -5.780e-01  2.193e-01  -2.636   0.0085 ** 
sulphates            8.652e-01  1.329e-01   6.509 1.09e-10 ***
alcohol              2.907e-01  3.040e-02   9.564  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6634 on 1267 degrees of freedom
Multiple R-squared:  0.3551,    Adjusted R-squared:  0.3495 
F-statistic: 63.41 on 11 and 1267 DF,  p-value: < 2.2e-16
```

The output shows the model parameters, residuals which are min-max values and quartiles and coefficients.It is possible to draw inferences about the significance of the model from the results obtained. The coefficient of denoted R^2, is the proportion of the variation in the dependent variable(target) that is predictable from the independent variables.(features) It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. At the same time, F- statistics and p-value are also among the values that we can look at to interpret the significance of the model. In this task ,we use the R^2 to measure the significance of the model. When we look at the R^2, it explains around the model 35% and we can tell that it is bad for the performance of the model.

5. Measuring Model Performance

After training model, we must check the model performance on test data. We can use the following metric (MSE,RMSE,MAE) to measure the performance of a regression model.

```
predicted_quality <- predict(lrm_model, test[,-12])
head(predicted_quality)
```

```
       1        2        3        4        5        6 
5.226765 5.116426 5.096636 5.368662 5.743560 5.542505 
```

To measure model performance, we first need to find the estimated value of the target variable. By the way, we should remove the target variable. After estimating the target variable, we need to calculate the error so that we can reach a conclusion about the performance of the model using the error.

```
error <- test$quality- predicted_quality
head(error)
```

```
          1           2           3           4           5           6 
-0.22676518 -0.11642590 -0.09663649 -0.36866196 -0.74355991 -0.54250544 
```

Measurements are based on the error values.

```
rmse_model <- sqrt(mean(error ^2))
rmse_model
```

```
[1] 0.5859451
```

We use some metrics when calculating the performance of the model. These are mean squared error, root mean squared error, and median absolute error. In this task we use the RMSE metric. Because when we look at MSE, it may be difficult to comment because its values will be large.

6. Checking the Possible Over and Underfitting Problem

In machine learning, there is an importing phenomenon which is called overfitting and underfitting. This is one of the main problem may faced in machine learning because we are training our models on train data then we measure the performance of the models on test data but we want to get a kind of model to be able to generalize to information that the model learned from train data to test data for doing this we should be avoid from the underfitting and overfitting problem. It was tested to see if there are any such problems in our model.

```r
rmse_train <- sqrt(mean((lrm_model$residuals) ^ 2))
rmse_test <- rmse_model
rmse_train - rmse_test
```

```
[1] 0.07435255
```

The output is positive. The differences is positive that the performance of the model is better on train set than test set. This may be a sign for underfitting problem because the model performance is better on train set. So, ıt is the insufficient learning of a model from the train set that cause the poorer performance on train set. How to solve underfitting problem? Training model with more features, training with more data, use more complex model, dealing with noise problem in data.

7. Predicting the Target Variable with the New Observation

A new observation was created and the value of the target feature was predicted.

```r
new_obs <- data.frame("redwine$fixed_acidity" = 8.3, "redwine$volatile_acidity" = 0.525,
                      "redwine$citric_acid" = 0.73,
                      "redwine$residual_sugar" = 3.80,
                      "redwine$chlorides" = 0.052, "redwine$free_sulfur_dioxide" = 35.0,
                      "redwine$total_sulfur_dioxide" = 123,
                      "redwine$density" = 0.99860, "redwine$pH" = 3.30,
                      "redwine$sulphates" = 1.24,
                      "redwine$alcohol" = 9.5)
```

```r
predicted_quality2 <- predict(lrm_model, new_data = new_obs)
head(predicted_quality2)
```

```
       1        2        3        4        5        6 
5.119390 6.385905 5.052443 5.422612 5.397811 5.683906 
```

Conclusion

The problem was identified in this assignment. Target variable and features were defined. The data set was examined and identified. The data set was divided into two. It's about to be a train and a test set. Linear regression models editorial. The performance measurements of the model were examined.(Root Mean Squared Error). It was decided that the performance of the model was poor. It was checked whether there were overfittig and underfitting Porblemes and it was concluded that there might be an underfitting problem in this model. Finally, a new observation was created and the value of the target feature was predicted.