**PROJECT #3**

**by**

**Dipika Mahashabde**

**(Cwid:803004274)**

**CPSC 483**

**Professor: Kenytt Avery**



**Department of Computer Science**

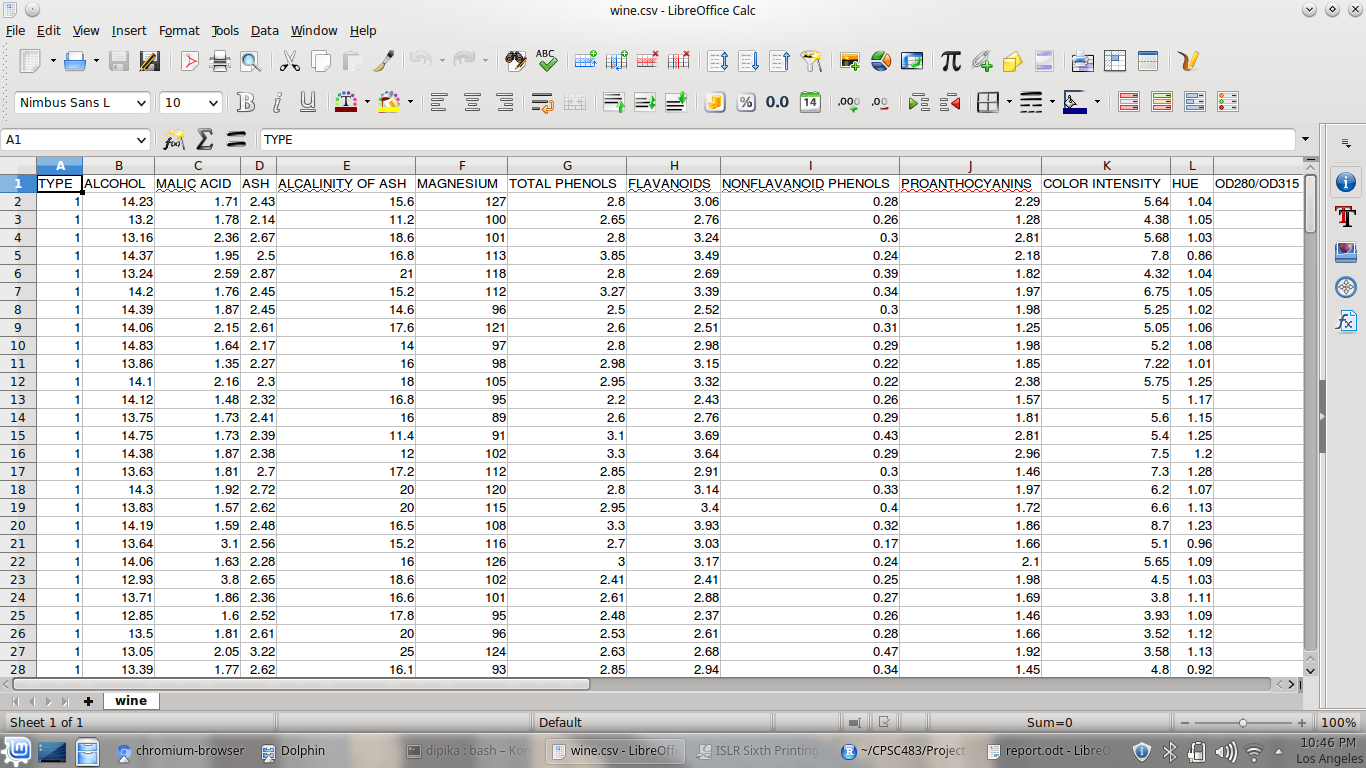
**California State University, Fullerton**

**Spring 2016**

**PART 1:- CHOOSING THE DATA SET**

URL: http://archive.ics.uci.edu/ml/machine-learning-databases/wine/

1. **The data I have chosen is Wine Data Set**



**PART 2 :- DATA MINING TASK:**

**CLASSIFICATION :-** This helps us to predict the type of wine it is whether it is type 1 or type 2 or type 3

**PART 3 :-CHOOSING A TASK**

1. features:**:-** the features I have chosen were dependent on my algorithms used. From these features I got maximum probabilities of correct prediction.

1. MALIC.ACID

2. ALCALINITY.OF.ASH

3. TOTAL.PHENOLS

4. NONFLAVANOID.PHENOLS

5. COLOR.INTENSITY

6. OD280.OD315.OF.DILUTED.WINES

**PART 4 :- SPLITTING INTO TEST DATA AND TRAINING DATA**

These lines of code will help us to split the data into test and training. I have divided it into 2/3

no is the no of rows in the data

boolean\_wine = sample.split(no,SplitRatio = 2/3,group = NULL)

Train.x=cbind(MALIC.ACID,MAGNESIUM,TOTAL.PHENOLS,NONFLAVANOID.PHENOLS,PROLINE,HUE)[boolean\_wine,]

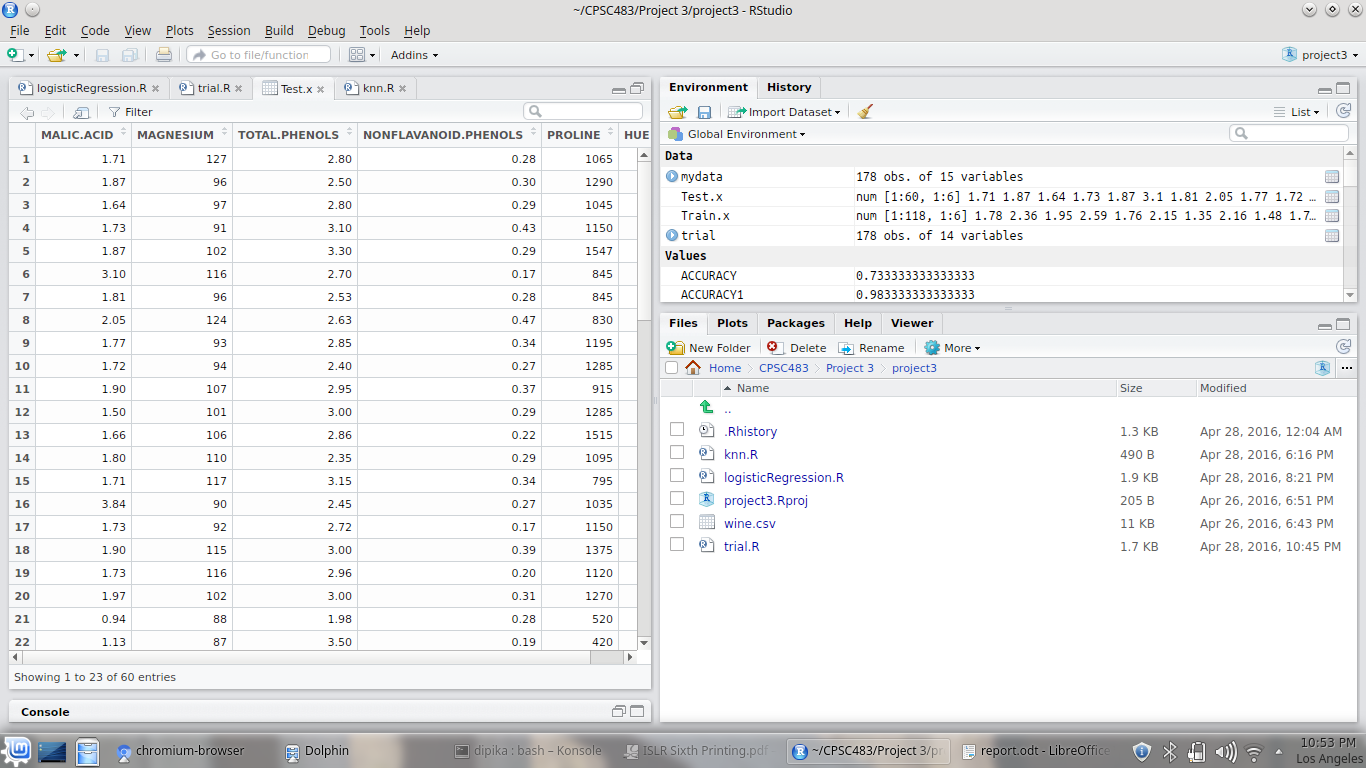
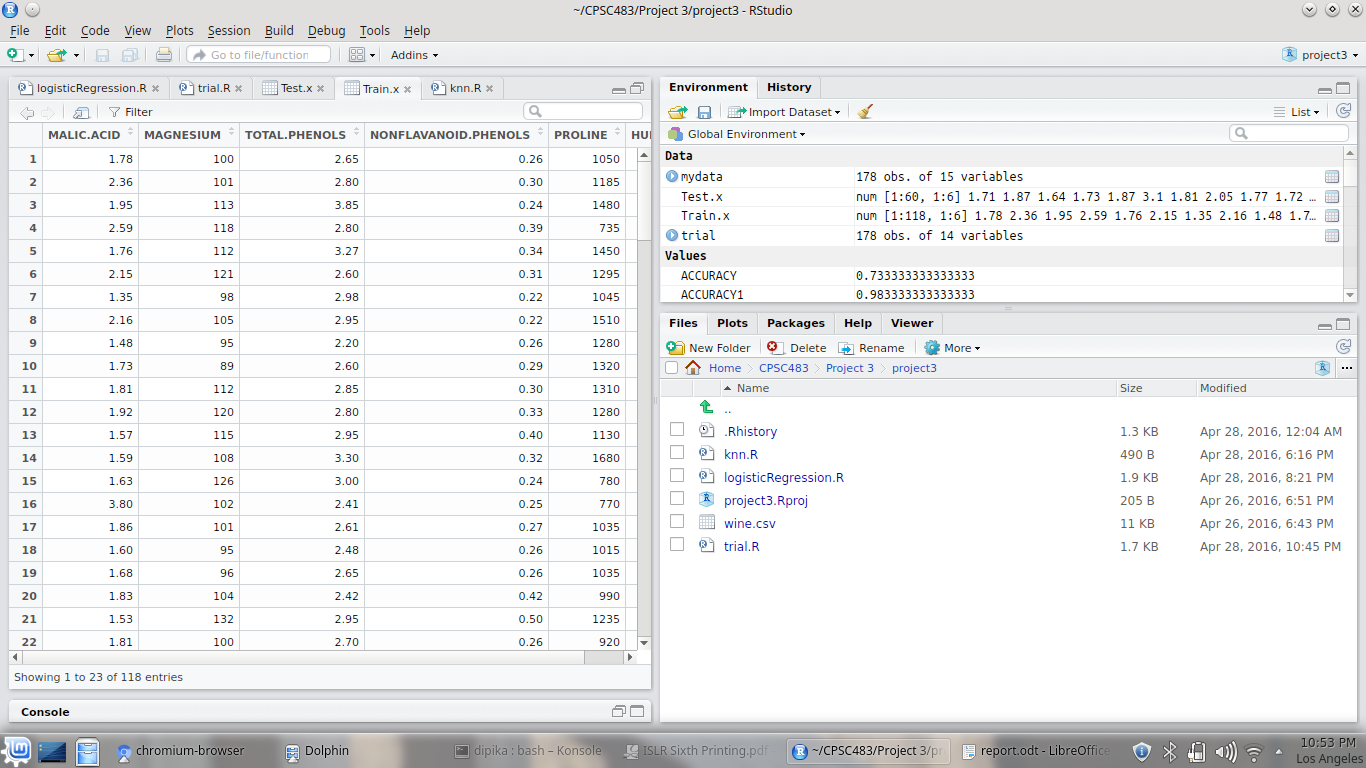
Test.x=cbind(MALIC.ACID,MAGNESIUM,TOTAL.PHENOLS,NONFLAVANOID.PHENOLS,PROLINE,HUE)[!boolean\_wine,]

Train.output=TYPE[boolean\_wine]

length(Train.output)

Test.output=TYPE[!boolean\_wine]

length(Test.output)



3. I have used **Logistic Regression** in that I have done two algorithm

1. General Logistic Regression and

2. Multinomial Logistic Regression.

Step1: In the General Logistic Regression I have manipulated the Type class into 3 parts, in the first part type 1=1 and other types=0

similarly,for other 2 types I have done the same.

After applying the glm function:-

glm.trial=glm((new\_trial)~MALIC.ACID+ALCALINITY.OF.ASH+TOTAL.PHENOLS+NONFLAVANOID.PHENOLS+COLOR.INTENSITY+OD280.OD315.OF.DILUTED.WINES,data=trial,family = binomial)

glm.trial1=glm((new\_trial\_type2)~MALIC.ACID+ALCALINITY.OF.ASH+TOTAL.PHENOLS+NONFLAVANOID.PHENOLS+COLOR.INTENSITY+OD280.OD315.OF.DILUTED.WINES,data=trial,family = binomial)

glm.trial2=glm(new\_trial\_type3~MALIC.ACID+ALCALINITY.OF.ASH+TOTAL.PHENOLS+NONFLAVANOID.PHENOLS+COLOR.INTENSITY+OD280.OD315.OF.DILUTED.WINES,data=trial,family = binomial)

glm.trial:-

Call:

glm(formula = (new\_trial) ~ MALIC.ACID + ALCALINITY.OF.ASH +

TOTAL.PHENOLS + NONFLAVANOID.PHENOLS + COLOR.INTENSITY +

OD280.OD315.OF.DILUTED.WINES, family = binomial, data = trial)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.95639 -0.26870 -0.06814 0.08536 2.67272

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -9.6519 3.4148 -2.826 0.00471 \*\*

MALIC.ACID 0.2010 0.3209 0.626 0.53104

ALCALINITY.OF.ASH -0.6865 0.1715 -4.003 6.25e-05 \*\*\*

TOTAL.PHENOLS 1.9269 0.7577 2.543 0.01099 \*

NONFLAVANOID.PHENOLS 3.8480 3.9426 0.976 0.32906

COLOR.INTENSITY 0.8482 0.1993 4.256 2.08e-05 \*\*\*

OD280.OD315.OF.DILUTED.WINES 4.0460 0.9762 4.145 3.40e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 226.13 on 177 degrees of freedom

Residual deviance: 74.17 on 171 degrees of freedom

AIC: 88.17

Number of Fisher Scoring iterations: 7

glm.trial1:-

Call:

glm(formula = (new\_trial\_type2) ~ MALIC.ACID + ALCALINITY.OF.ASH +

TOTAL.PHENOLS + NONFLAVANOID.PHENOLS + COLOR.INTENSITY +

OD280.OD315.OF.DILUTED.WINES, family = binomial, data = trial)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.49753 -0.18293 -0.00661 0.19766 2.27201

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 6.6422 3.1261 2.125 0.03361 \*

MALIC.ACID -1.0769 0.3310 -3.254 0.00114 \*\*

ALCALINITY.OF.ASH 0.2283 0.1272 1.796 0.07255 .

TOTAL.PHENOLS 0.8148 0.7632 1.068 0.28572

NONFLAVANOID.PHENOLS 3.2408 3.3426 0.970 0.33227

COLOR.INTENSITY -2.3441 0.4110 -5.703 1.18e-08 \*\*\*

OD280.OD315.OF.DILUTED.WINES -0.7324 0.6520 -1.123 0.26129

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 239.429 on 177 degrees of freedom

Residual deviance: 72.682 on 171 degrees of freedom

AIC: 86.682

Number of Fisher Scoring iterations: 7

> summary ( glm.trial2 )

Call:

glm(formula = new\_trial\_type3 ~ MALIC.ACID + ALCALINITY.OF.ASH +

TOTAL.PHENOLS + NONFLAVANOID.PHENOLS + COLOR.INTENSITY +

OD280.OD315.OF.DILUTED.WINES, family = binomial, data = trial)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.51505 -0.02280 -0.00373 0.00014 1.38644

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -6.0980 8.1586 -0.747 0.4548

MALIC.ACID 1.2928 0.7569 1.708 0.0876 .

ALCALINITY.OF.ASH 0.6070 0.4301 1.411 0.1581

TOTAL.PHENOLS -3.2789 2.0227 -1.621 0.1050

NONFLAVANOID.PHENOLS -5.3769 7.2504 -0.742 0.4583

COLOR.INTENSITY 2.1946 1.0175 2.157 0.0310 \*

OD280.OD315.OF.DILUTED.WINES -5.6088 2.4471 -2.292 0.0219 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 207.52 on 177 degrees of freedom

Residual deviance: 15.32 on 171 degrees of freedom

AIC: 29.32

Number of Fisher Scoring iterations: 10

Step 3. After this I removed the probabilites of these three glm function

T1=predict(glm.trial,trial,type = "response")

T2=predict(glm.trial1,trial,type = "response")

T3=predict(glm.trial2,trial,type="response")

took the weighted\_mean of the three got the probability of getting the output .

1. Multinomial Logistic Regression

I have got trained the train data using multinomial function

mymodel<-multinom(TYPE~MALIC.ACID+MAGNESIUM+TOTAL.PHENOLS+NONFLAVANOID.PHENOLS+PROLINE+HUE,data=mydata)

summary(mymodel)

from this I applied this trained model to my test data

test\_predict=predict(mymodel,Test.x,"probs")

test\_predict

form that I compared them with the actual outputs

> print(table1)

Test.output

test\_predict 1 2 3

1 20 1 0

2 0 23 0

3 0 0 16

from this I calculated the accuracy precision and recall of the model

recall1=(table1[1,1]/(sum(table1[1,])))

recall2=(table1[2,2]/(sum(table1[2,])))

recall3=(table1[3,3]/(sum(table1[3,])))

recall\_logistic=(recall3+recall2+recall1)/3

recall\_logistic:-> recall\_logistic

[1] 0.984127

prec\_1=(table1[1,1]/(sum(table1[,1])))

prec\_2=(table1[2,2]/(sum(table1[,2])))

prec\_3=(table1[3,3]/(sum(table1[,3])))

prec\_logistic=(prec\_3+prec\_2+prec\_1)/3

prec\_logistic:-

> prec\_logistic

[1] 0.9861111

ACCURACY1=sum(diag(table1))/sum(table1)

ACCURACY1 :- 98%

PART 4 - EXPLORING ALTERNATIVES

For the Exploring Alternatives I have used KNN(k-nearest neighbors)

the same Training and test data I have used to compute KNN.

set.seed(1)

library(class)

knn.pred=knn(Train.x,Test.x,Train.output,k=9)

test.class=TYPE[!boolean\_wine]

table=table(knn.pred,test.class)

table

ACCURACY=(sum(diag(table)))/sum(table)

ACCURACY

recall1=(table[1,1]/(sum(table[1,])))

recall1

recall2=(table[2,2]/(sum(table[2,])))

recall3=(table[3,3]/(sum(table[3,])))

recall=(recall3+recall2+recall1)/3

recall

prec1=(table[1,1]/(sum(table[,1])))

prec2=(table[2,2]/(sum(table[,2])))

prec3=(table[3,3]/(sum(table[,3])))

prec=(prec1+prec2+prec3)/3

prec

the K value used in the knn function is to check with the k nearest points and take the highest pointed value. Mostly the k value should be a odd number so there wont be a tie.

I tried with different values for K started from 3-31

the accuracy kept changing from 3-9 it increased but form 9onwards it started to decrease so K=9 was perfect value for this function

> ACCURACY

[1] 0.7333333

> recall

[1] 0.7326045

> prec

[1] 0.725

RESULTS

Therefore after comparing the two algorithm used first Multinomial Logistic Regression and second KNN

We see that the Multinomial Logistic Regression gives higher probability of correct answers

REFERENCES:

1. <http://www.inside-r.org/r-doc/stats/glm>
2. <http://www.ats.ucla.edu/stat/r/dae/mlogit.html>
3. An introduction to Statistical Learning