Student's Analysis Report

Table of Contents

S.NO	Title	Page No.
1.	Abstract	3
2.	Introduction	4
3.	Dataset Description	5
4.	Algorithms	6
5.	Implementation	7
6.	Conclusion	17
7.	References	18

1. Abstract

Although the educational level of the Portuguese population has improved in the last decades, the statistics keep Portugal at Europe's tail end due to its high student failure rates. In particular, lack of success in the core classes of Mathematics and the Portuguese language is extremely serious. On the other hand, the fields of Business Intelligence (BI)/Data Mining (DM), which aim at extracting high-level knowledge from raw data, offer interesting automated tools that can aid the education domain. Recent real-world data (e.g. student grades, demographic, social and school related features) was collected by using school reports and questionnaires. The two core classes (i.e. Mathematics and Portuguese) were modeled under binary/five-level classification and regression tasks. The results show that a good predictive accuracy can be achieved, provided that the first and/or second school period grades are available. Although student achievement is highly influenced by past evaluations, an explanatory analysis has shown that there are also other relevant features (e.g. number of absences, parent's job and education, alcohol consumption). As a direct outcome of this project, more efficient student prediction tools can be developed, improving the quality of education and enhancing school resource management.

2. Introduction

Education is a key factor for achieving a long-term economic progress. During the last decades, the Portuguese educational level has improved. However, the statistics keep the Portugal at Europe's tail end due to its high student failure and dropping out rates. For example, in 2006 the early school leaving rate in Portugal was 40% for 18 to 24 year olds, while the European Union average value was just 15% (Eurostat 2007). In particular, failure in the core classes of Mathematics and Portuguese (the native language) is extremely serious, since they provide fundamental knowledge for the success in the remaining school subjects (e.g. physics or history). On the other hand, the interest in Business Intelligence (BI)/Data Mining (DM) (Turban et al. 2007), arose due to the advances of Information Technology, leading to an exponential growth of business and organizational databases. All this data holds valuable information, such as trends and patterns, which can be used to improve decision making and optimize success. Yet, human experts are limited and may overlook important details. Hence, the alternative is to use automated tools to analyze the raw data and extract interesting high level information for the decision-maker. The education arena offers a fertile ground for BI applications, since there are multiple sources of data (e.g. traditional databases, online web pages) and diverse interest groups (e.g. students, teachers, administrators or alumni) (Ma et al. 2000). Who is likely to return for more classes? What type of courses can be offered to attract more students? What are the main reasons for student transfers? Is it possible to predict student performance? What are the factors that affect student achievement? Modeling student performance is an important tool for both educators and students, since it can help a better understanding of this phenomenon and ultimately improve it. For instance, school professionals could perform corrective measures for weak students

(e.g. remedial classes).

3. Dataset Description

In Portugal, the secondary education consists of 3 years of schooling, preceding 9 years of basic education and followed by higher education. Most of the students join the public and free education system. There are several courses (e.g. Sciences and Technologies, Visual Arts) that share core subjects such as the Portuguese Language and Mathematics. Like several other countries (e.g. France or Venezuela), a 20-point grading scale is used, where 0 is the lowest grade and 20 is the perfect score. During the school year, students are evaluated in three periods and the last evaluation (G3 of Table 1) corresponds to the final grade. This study will consider data collected during the 2005- 2006 school year from two public schools, from the Alentejo region of Portugal. Although there has been a trend for an increase of Information Technology investment from the Government, the majority of the Portuguese public school information systems are very poor, relying mostly on paper sheets (which was the current case). Hence, the database was built from two sources: school reports, based on paper sheets and including few attributes (i.e. the three period grades and number of school absences); and questionnaires, used to complement the previous information. We designed the latter with closed questions (i.e. with predefined options) related to several demographic (e.g. mother's education, family income), social/emotional (e.g. alcohol consumption) (Pritchard and Wilson 2003) and school related (e.g. number of past class failures) variables that were expected to affect student performance. The questionnaire was reviewed by school professionals and tested on a small set of 15 students in order to get a feedback. The final version contained 37 questions in a single A4 sheet and it was answered in class by 788 students. Latter, 111 answers were discarded due to lack of identification details (necessary for merging with the school reports). Finally, the data was integrated into two datasets related to Mathematics (with 395 examples) and the Portuguese language (649 records) classes.

During the preprocessing stage, some features were discarded due to the lack of discriminative value. For instance, few respondents answered about their family income (probably due to privacy issues), while almost 100% of the students live with their parents and have a personal computer at home. The remaining attributes are shown in Table 1, where the last four rows denote the variables taken from the school reports.

Table 1: The preprocessed student related variables

Attribute	Description (Domain)
sex	student's sex (binary: female or male)
age	student's age (numeric: from 15 to 22)
school	student's school (binary: *Gabriel Pereira* or *Mousinho da Silveira*)
address	student's home address type (binary: urban or rural)
Pstatus	parent's cohabitation status (binary: living together or apart)
Medu	mother's education (numeric: from 0 to 4[a])
Mjob	mother's job (nominal[b])
Fedu	father's education (numeric: from 0 to 4[a])
Fjob	father's job (nominal[b])
guardian	student's guardian (nominal: mother, father or other)
famsize	family size (binary: ≤ 3 or > 3)
famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
reason	reason to choose this school (nominal: close to home, school reputation, course preference or other)
traveltime	home to school travel time (numeric: 1 – $<$ 15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour or 4 – $>$ 1 hour).
studytime	weekly study time (numeric: 1 – $<$ 2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours or 4 – $>$ 10 hours)
failures	number of past class failures (numeric: n if $1 \leq n < 3$, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
freetime	free time after school (numeric: from 1 – very low to 5 – very high)
goout	going out with friends (numeric: from 1 – very low to 5 – very high)
Walc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Dalc	workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
health	current health status (numeric: from 1 – very bad to 5 – very good)
absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20)

a 0 – none, 1 – primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education.
b teacher, health care related, civil services (e.g. administrative or police), at home or other.

4. Algorithms

a) Linear Regression

b) Lasso Regression

c) Ridge Regression

d) Decision Tree Regression

e) Decision Tree Classifier

f) Random Forest Classifier

g) Ada Boosting Classifier

h) K- Nearest Neighbor

i) Logistic Regression

j) Gaussian Naïve Bayes

k) Bernoulli Naïve Bayes

5. Implementation

a) First 10 rows of Data with AvgGrade[G1 G2 G3]

```
df=pd.read_csv('F:\\VIT SEMESTER 6\\Machine Learning\\projects\\alcohol consumption\\student-alcohol-consumption\\student-mat.csv
```

```
df.head(10)
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	AvgGrade
0	2	2	18	2	2	1	4	4	4	5	...	3	4	1	1	3	6	5	6	6	5.666667
1	2	2	17	2	2	2	1	1	4	3	...	3	3	1	1	3	4	5	5	6	5.333333
2	2	2	15	2	1	2	1	1	4	3	...	3	2	2	3	3	10	7	8	10	8.333333
3	2	2	15	2	2	2	4	2	2	1	...	2	2	1	1	5	2	15	14	15	14.666667
4	2	2	16	2	2	2	3	3	3	3	...	3	2	1	2	5	4	6	10	10	8.666667
5	2	1	16	2	1	2	4	3	1	3	...	4	2	1	2	5	10	15	15	15	15.000000
6	2	1	16	2	1	2	2	2	3	3	...	4	4	1	1	3	0	12	12	11	11.666667
7	2	2	17	2	2	1	4	4	3	5	...	1	4	1	1	1	6	6	5	6	5.666667
8	2	1	15	2	1	1	3	2	1	3	...	2	2	1	1	1	0	16	18	19	17.666667
9	2	1	15	2	2	2	3	4	3	3	...	5	1	1	1	5	0	14	15	15	14.666667

10 rows × 34 columns

b) Pairplot between ‘Studytime’,’failure’, ‘G1’

```
sns.pairplot(df[['studytime','failures','G1']], size=3);
```

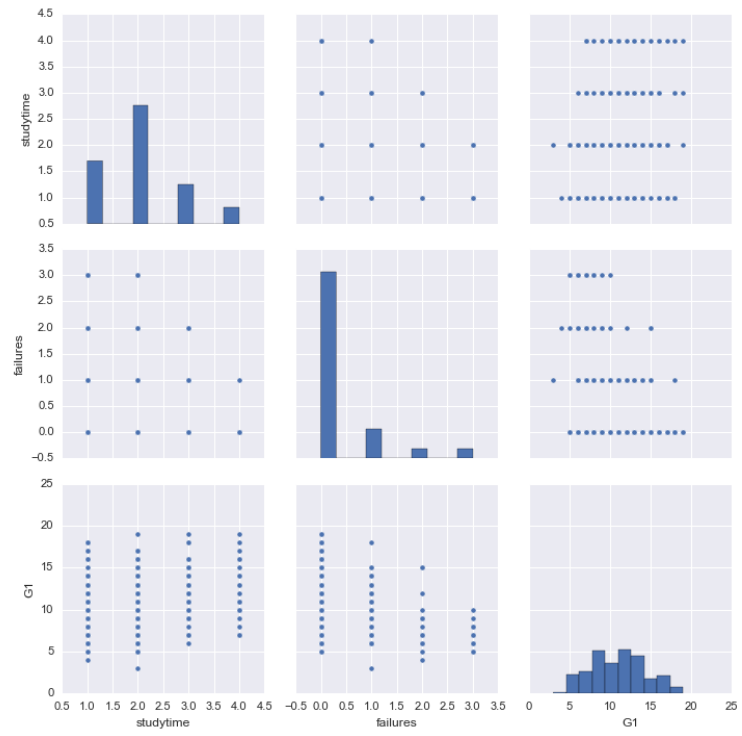

Fig 1. Pairplot between [‘studytime’,’failure’,’G1’]

c) Student distribution in two schools according to gender

```
b = sns.factorplot(x="school", y="age", hue="sex", data=df, kind="bar", palette="muted" )
```

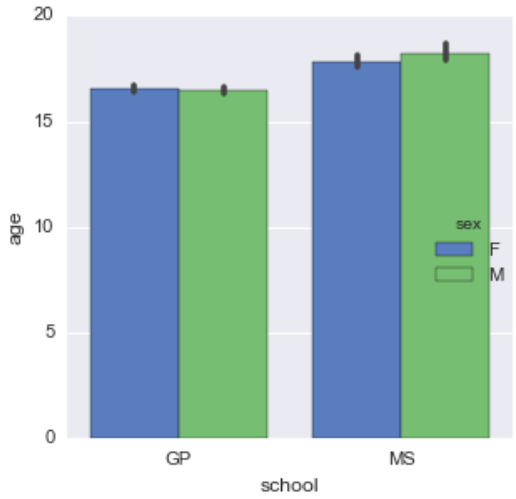

Fig 2. Histogram of school and age

d) Linear Regression Output

```
predicted = cross_val_predict(regr, X_test, y_test, cv=10)

fig, ax = plt.subplots()
ax.scatter(y_test, predicted)
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Measured')
ax.set_ylabel('Predicted')
plt.show()
```

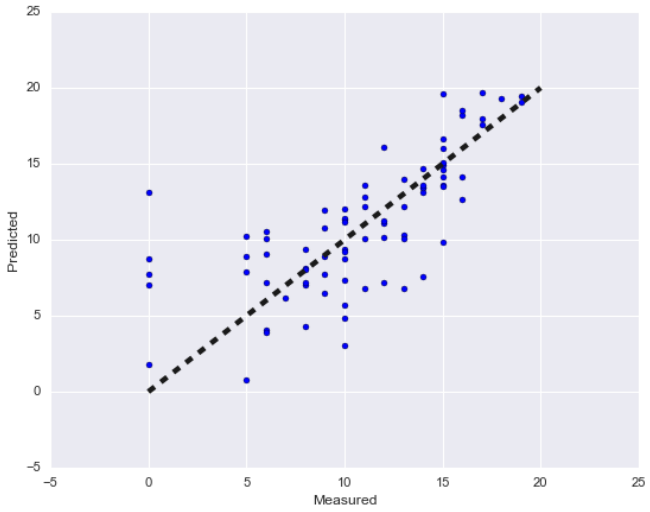

Fig 3. Linear Regression

e) Study Time Distribution of students

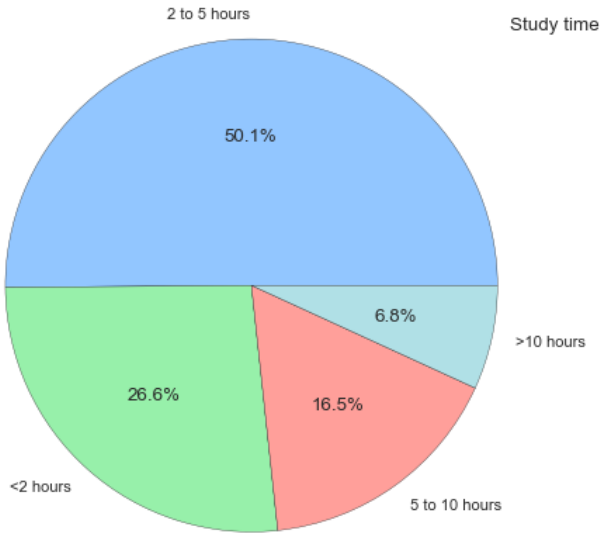

Fig 4. Pie Chart of student study time

f) Weekend and workday alcohol consumption with respect to Health

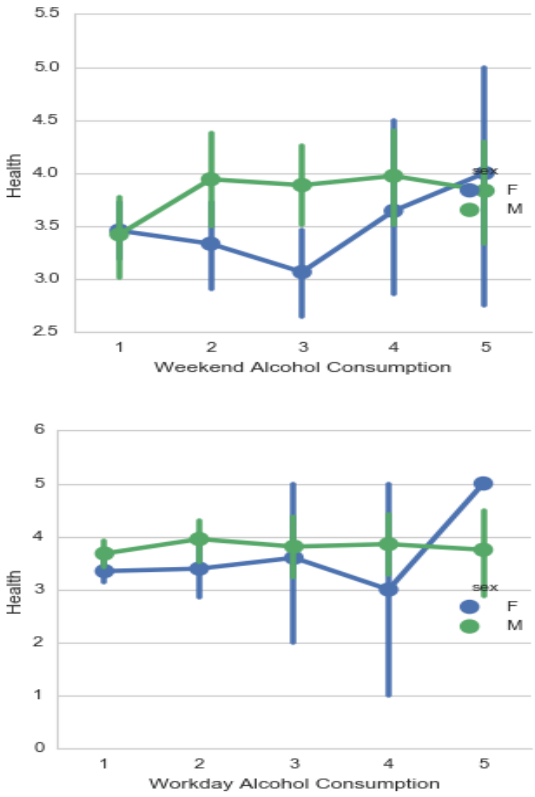

Fig 5. Factorplot of alcohol Consumption

g) Final grade with respect to weekday and weekend alcohol consumption

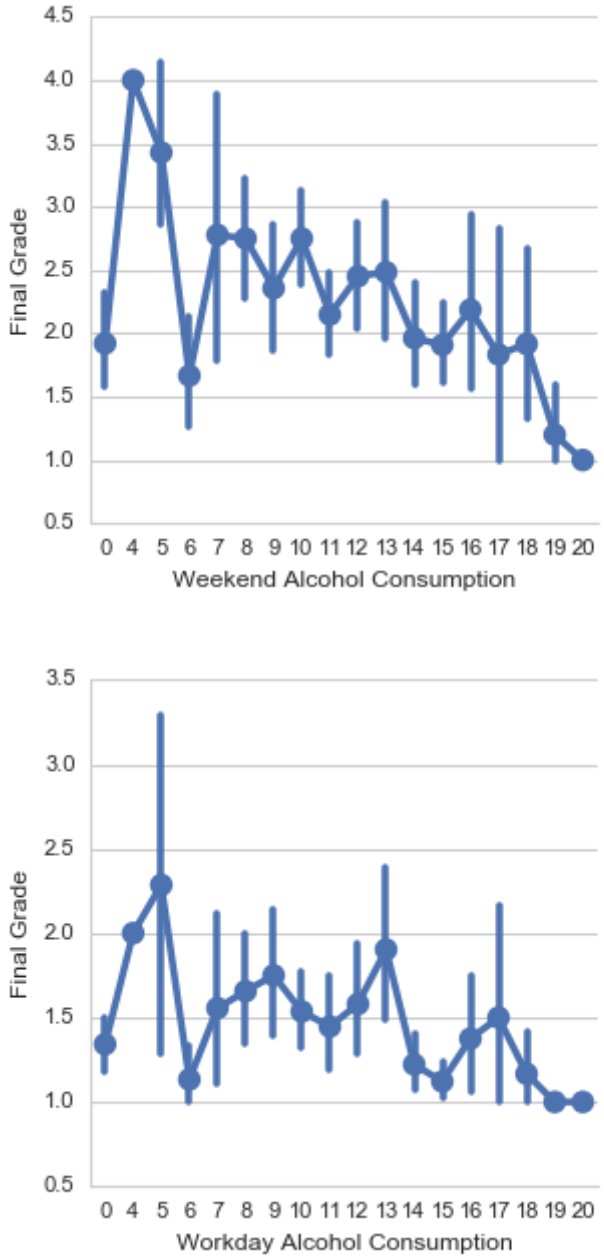

Fig 6. Factorplot between FinalGrade and alcohol consumption

h) Decision Tree, Lasso Regression and Ridge Regression

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.linear_model import Ridge

from sklearn.model_selection import cross_val_score
```

```
y = df['G3']
X = df.drop(['G3'], axis=1)
```

```
X = pd.get_dummies(X)
```

```
names = ['DecisionTreeRegressor', 'LinearRegression', 'Ridge', 'Lasso']

clf_list = [DecisionTreeRegressor(),
            LinearRegression(),
            Ridge(),
            Lasso()]
```

```
for name, clf in zip(names, clf_list):
    print name, ':',cross_val_score(clf, X, y, cv=5).mean()
```

```
DecisionTreeRegressor : 0.949299130716
LinearRegression : 1.0
Ridge : 0.999964917992
Lasso : 0.910358876299
```

Fig 7. Accuracy of corresponding algorithms

i) Decision Tree Classifier

```
yy=[str(i) for i in y]
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, yy, random_state = 0)
clf = DecisionTreeClassifier().fit(X_train, y_train)
# list(set(yy))
print(clf.score(X_train,y_train))
print(clf.score(X_test,y_test))
```

```
1.0
0.545454545455
```

Fig 8. Decision Tree Classifier Accuracy on training and testing data

j) Decision Tree

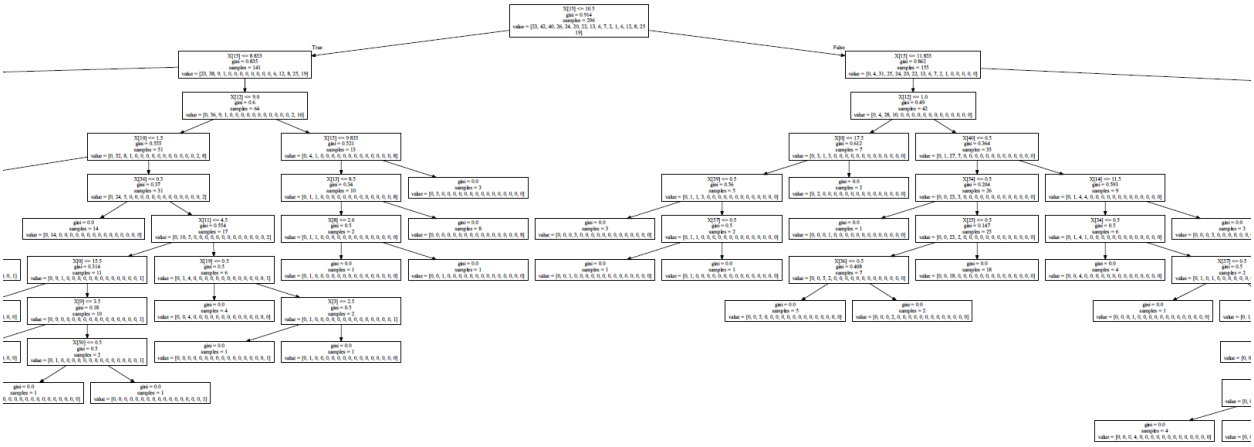

Fig 9. Decision Tree Center

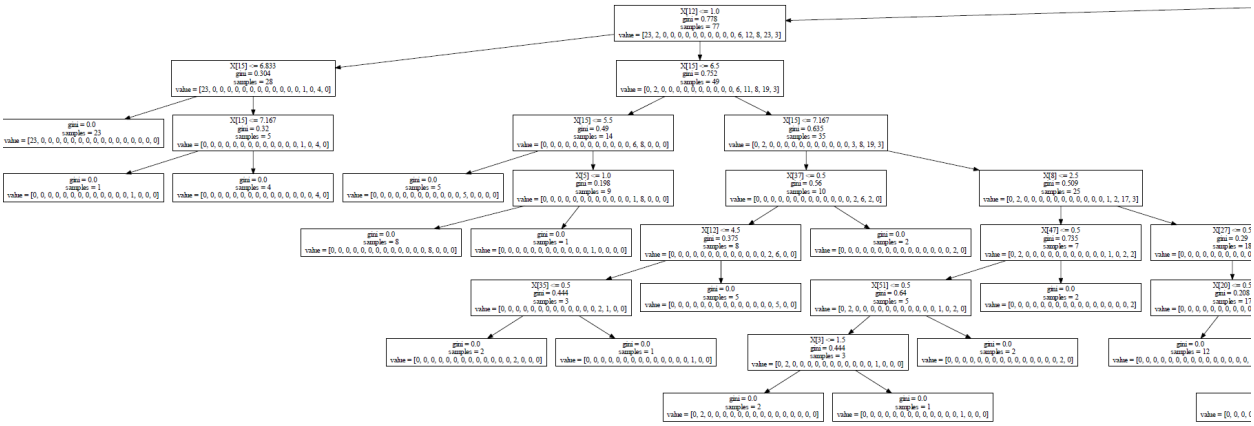

Fig 10. Decision Tree Left Half

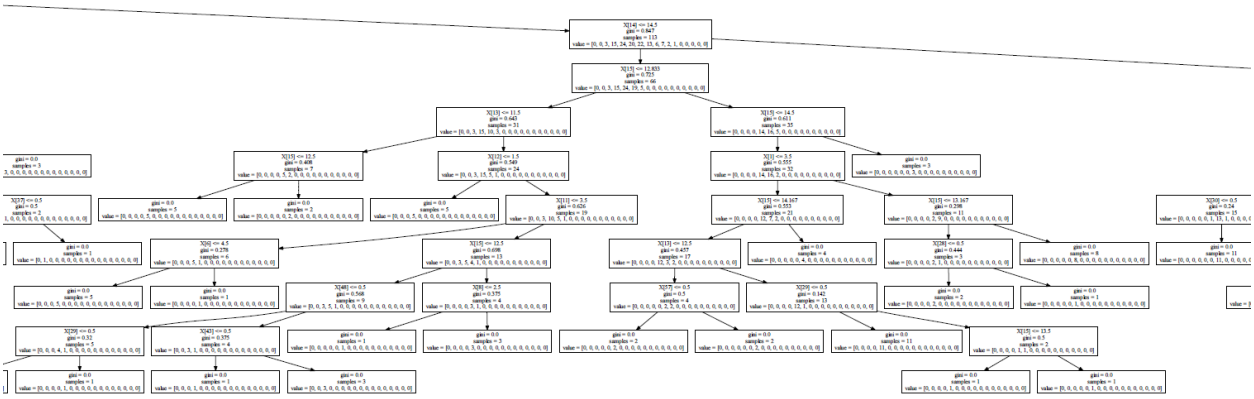

Fig 11. Decision Tree Right Half

k) Heat map of correlation between all attributes

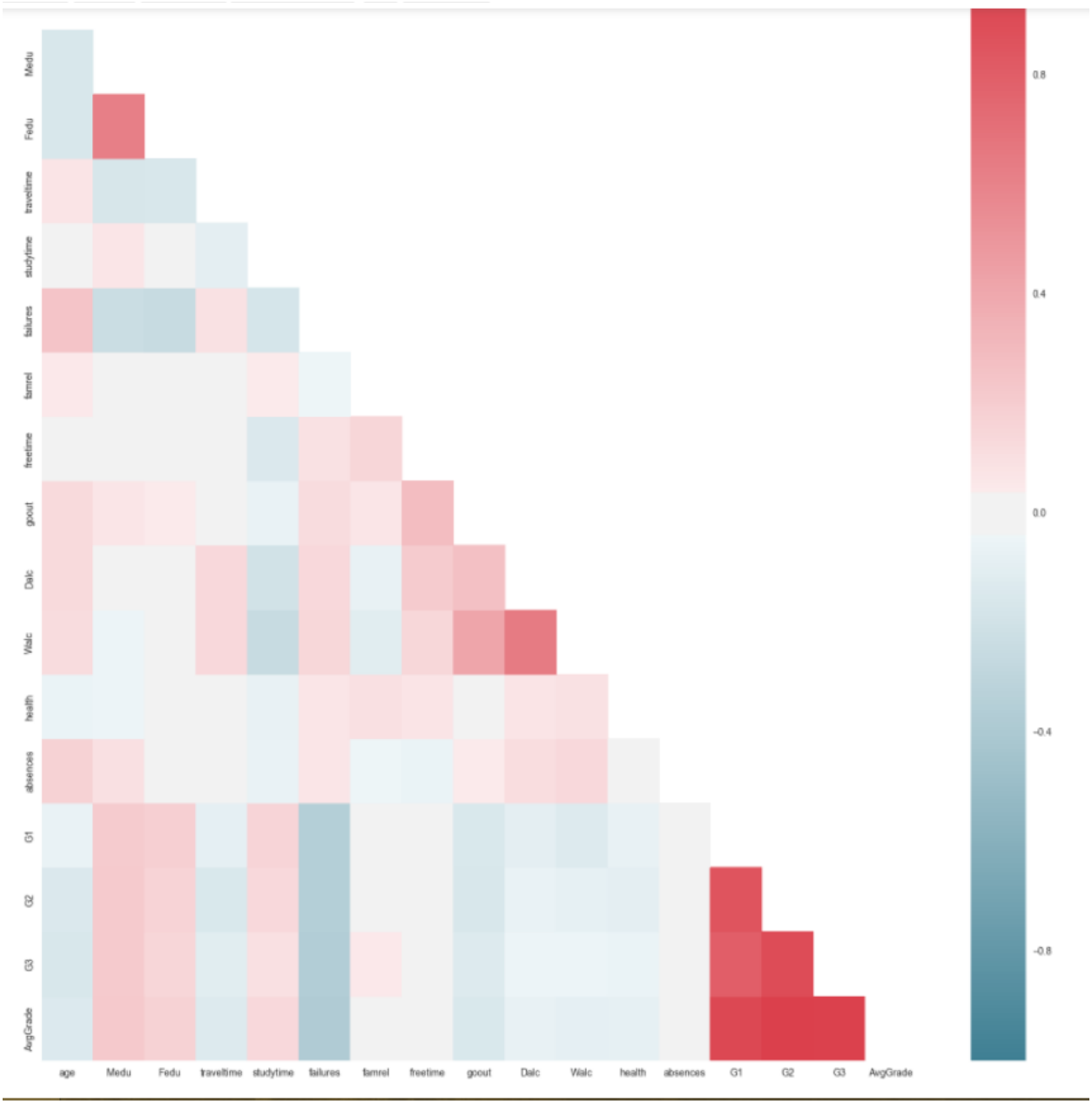

Fig 12. Correlation Heat Map

I) Random Forest Classifier to find feature imnportance

```
from sklearn.linear_model import SGDClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.calibration import CalibratedClassifierCV
#import xgboost as xgb
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import LogisticRegression
from sklearn import svm

df_copy = pd.get_dummies(mod_df)

df1 = df_copy
y = np.asarray(df1['AvgGrade'], dtype="|S6")
df1 = df1.drop(['AvgGrade'],axis=1)
X = df1.values
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.50)

radm = RandomForestClassifier()
radm.fit(Xtrain, ytrain)

clf = radm
indices = np.argsort(radm.feature_importances_)[::-1]

# Print the feature ranking
print('Feature ranking:')

for f in range(df1.shape[1]):
    print('%d. feature %d %s (%f)' % (f+1 ,
                                      indices[f],
                                      df1.columns[indices[f]],
                                      radm.feature_importances_[indices[f]]))
```

```
Feature ranking:
1. feature 30 G1 (0.088482)
2. feature 31 G2 (0.087863)
3. feature 32 G3 (0.081168)
4. feature 29 absences (0.062547)
5. feature 2 age (0.050030)
6. feature 23 famrel (0.045402)
7. feature 25 goout (0.042331)
8. feature 6 Medu (0.039715)
9. feature 28 health (0.035614)
10. feature 24 freetime (0.034163)
11. feature 27 Walc (0.034087)
12. feature 8 Mjob (0.033830)
13. feature 10 reason (0.033398)
14. feature 7 Fedu (0.032304)
15. feature 9 Fjob (0.026092)
16. feature 13 studytime (0.024720)
17. feature 12 traveltime (0.023986)
18. feature 4 famsize (0.020248)
19. feature 22 romantic (0.018892)
20. feature 14 failures (0.018447)
21. feature 16 famsup (0.017781)
22. feature 21 internet (0.017760)
23. feature 26 Dalc (0.017623)
24. feature 17 paid (0.016871)
25. feature 18 activities (0.014304)
26. feature 19 nursery (0.014201)
27. feature 5 Pstatus (0.013003)
28. feature 1 sex (0.012433)
29. feature 3 address (0.011195)
30. feature 20 higher (0.009584)
31. feature 15 schoolsup (0.008001)
32. feature 11 guardian (0.007435)
33. feature 0 school (0.006491)
```

Fig 13. Feature Ranking

m) Random Forest(G,E), Ada Boosting, Extra Tree Classifier, K Neighbors, Decision Tree, Logistic Regression, Gaussian Naïve Bayes, Bernoulli Naïve Bayes

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.decomposition import PCA
from sklearn.model_selection import cross_val_score
from sklearn.feature_selection import RFECV, SelectKBest
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB, BernoulliNB
from sklearn.neighbors import KNeighborsClassifier

classifiers = [('RandomForestClassifierG', RandomForestClassifier(n_jobs=-1, criterion='gini')),
               ('RandomForestClassifierE', RandomForestClassifier(n_jobs=-1, criterion='entropy')),
               ('AdaBoostClassifier', AdaBoostClassifier()),
               ('ExtraTreesClassifier', ExtraTreesClassifier(n_jobs=-1)),
               ('KNeighborsClassifier', KNeighborsClassifier(n_jobs=-1)),
               ('DecisionTreeClassifier', DecisionTreeClassifier()),
               ('ExtraTreeClassifier', ExtraTreeClassifier()),
               ('LogisticRegression', LogisticRegression()),
               ('GaussianNB', GaussianNB()),
               ('BernoulliNB', BernoulliNB()),
              ]
allscores = []

x, Y = mod_df.drop('AvgGrade', axis=1), np.asarray(mod_df['AvgGrade'], dtype="|S6")

for name, classifier in classifiers:
    scores = []
    for i in range(20): # 20 runs
        roc = cross_val_score(classifier, x, Y)
        scores.extend(list(roc))
    scores = np.array(scores)
    print(name, scores.mean())
    new_data = [(name, score) for score in scores]
    allscores.extend(new_data)
```

```
('RandomForestClassifierG', 0.17442515959108967)
('RandomForestClassifierE', 0.155050220972278)
('AdaBoostClassifier', 0.09382728449622782)
('ExtraTreesClassifier', 0.1435033257443864)
('KNeighborsClassifier', 0.11395741261550821)
('DecisionTreeClassifier', 0.28300352662827549)
('ExtraTreeClassifier', 0.099803133788670145)
('LogisticRegression', 0.10047587161287441)
('GaussianNB', 0.11942591848578189)
('BernoulliNB', 0.053892236953707415)
```

Fig 14. Accuracy of Algorithms

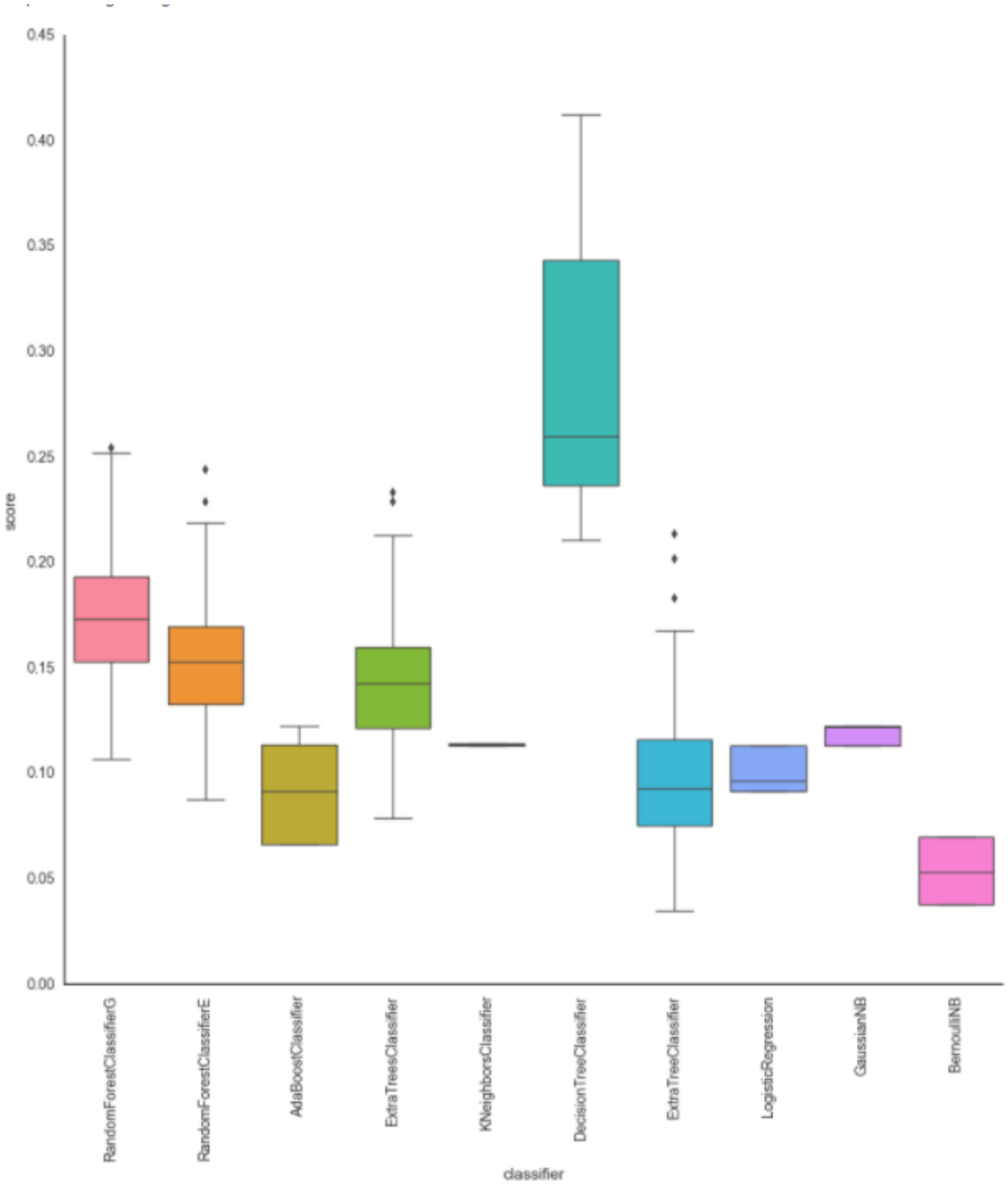

Fig 15. Box plot of accuracies of all algorithms

6. Conclusion

Based on the results we can see that in decision tree regressor the most important features are avggrade, absences, G1, G2, health, age, freetime, goout, studytime. The rest of the features are of mere importance with importance value less than 0.005. The accuracy achieved by decision tree for training data is 100% and testing data is 54.54% which means that overfitting of data is occurring. The training accuracy achieved by lasso, linear regression, ridge regression 91.03%, 100%, 99.99% which is also showing overfitting of data.

The accuracy achieved by other algorithms are:

1. RandomForestClassifier (Gain): 17.44%
2. RandomForestClassifier(Entropy): 15.55%
3. AdaBoostClassifier: 9.38%
4. ExtraTreesClassifier: 14.35%
5. KNeighborsClassifier: 11.39%
6. DecisionTreeClassifier: 28.30%
7. LogisticRegression: 10.04%
8. Gaussian Naïve Bayes: 11.94%
9. Bernoulli Naïve Bayes: 5.38%

We have addressed the prediction of secondary student grades of two core classes (Mathematics and Portuguese) by using past school grades (first and second periods), demographic, social and other school related data. The data was tested using several methods of machine learning. Also, distinct input selections (e.g. with or without past grades) were explored. The obtained results reveal that it is possible to achieve a predictive accuracy, provided that the first and/or second school period grades are known. This confirms that student achievement is highly affected by previous performances.

Nevertheless, an analysis to knowledge provided by the best predictive models has shown that, in some cases, there are other relevant features, such as: school related (e.g. number of absences, reason to choose school, extra educational school support), demographic (e.g. student's age, parent's job and education) and social (e.g. going out with friends, alcohol consumption) variables. This study was based on an off-line learning, since the DM techniques were applied after the data was collected. However, there is a potential for an automatic on-line learning environment, by using a student prediction engine as part of a school management support system. This will allow the collection of additional features (e.g. grades from previous school years) and also to obtain a valuable feedback from the school professionals. Furthermore, we intent to enlarge the experiments to more schools and school years, in order to enrich the student databases. More research is also needed (e.g. sociological studies) in order to understand why and how some variables (e.g. reason to choose school, parent's job or alcohol consumption) affect student performance.

7. References

1. https://archive.ics.uci.edu/ml/datasets/student+performance
2. http://scikit-learn.org/stable/
3. http://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifie.html
4. http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
5. http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
6. http://scikitlearn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
7. http://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
8. http://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
9. http://scikitlearn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html