PREDICTING THE STUDENT PERFORMANCE BASED ON EXTERNAL AND INTERNAL CRITERIA/FEATURES FOR ASSESSING A BINARY OUTCOME

By: SHREY SHARMA, RITWICK DEV

Royal Melbourne Institute of Technology

Contact Details –

Shrey Sharma - *s3696695@student.rmit.edu.au*

Ritwick Dev - *s3702041@student.rmit.edu.au*

27th May ,2019

TABLE OF CONTENT

ABSTRACT

The Idea behind the accession of the dataset was to purely analyse the effect of various factors on the performance index of the individual. Portuguese high school performances have been significantly increasing in the last decade. However, their performance is still considered to be below par in comparisons to other European countries like England, France. A high amount of failure rate was prevalent specially in subjects like Maths and Portuguese, which is an alarming concern for the Department of Education in Portugal. By using, significant data exploration and using machine learning algorithms like KNN Classification and Decision tree, we'll try to dig deep, analyse raw data and try to predict student performance with respect to their highly correlated features like past exam records, their social life performance, parent's status, relationship with other mates, interest in studies etc. Later, we'll compare all the produced models based on different parameters to find the best model for this dataset.

INTRODUCTION

The dataset is from UCI machine learning repository.

We have worked on "student-mat.csv"dataset(only Maths Subject) taken from 'Data Folder' provided in the website.

This dataset consists of 33 columns with 395 observations mostly with categorical and binary features. This dataset has features related with student's social and academic information. We'll be predicting student's academic performance based on the target feature 'G3' (final grade scored in Maths subject).

We'll be analysing and predicting student's performance for two of the Portuguese high schools.

Attribute Information –

- **school:** student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
- **sex:** student's sex (binary: 'F' - female or 'M' - male)
- **age:** student's age (numeric: from 15 to 22)
- **address:** student's home address type (binary: 'U' - urban or 'R' - rural)
- **famsize:** family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
- **Pstatus:** parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
- **Medu:** mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- **Fedu:** father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- **Mjob:** mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

- **Fjob:** father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
- **reason:** reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
- **guardian:** student's guardian (nominal: 'mother', 'father' 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<=n<3, else 4)
- **schoolsup:** extra educational support (binary: yes or no)
- **famsup:** family educational support (binary: yes or no)
- **paid:** extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- **activities:** extra-curricular activities (binary: yes or no)
- **nursery:** attended nursery school (binary: yes or no)
- **higher:** wants to take higher education (binary: yes or no)
- **internet:** Internet access at home (binary: yes or no)
- **romantic:** with a romantic relationship (binary: yes or no)
- **famrel:** quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- **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)
- **Dalc:** workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- **Walc:** weekend 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).

METHODOLOGY

Before doing data exploration, we pre-processed the whole dataset to check the presence of NULL values. Fortunately, there were no NULL values present. Checked range of all the integer related attribute. As the range were not that huge, we decided not to apply feature scaling for any attributes.

- **Univariate Analysis**
  Selected and explored 10 best explained attributes by using pie chart and histogram plot. Here are couple of univariate plots that we explored.

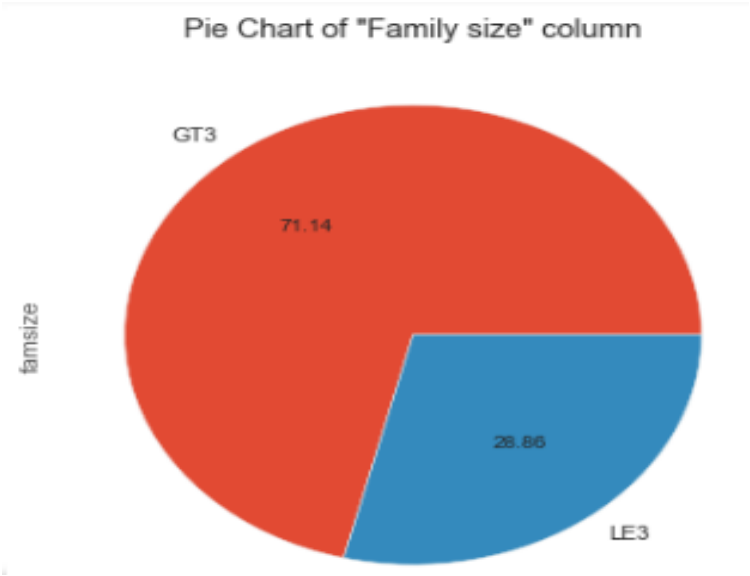

Fig I

The above Fig I. pie chart shows that there are almost 71% of students with family size greater than 3 while only 27% have family size less than 3.

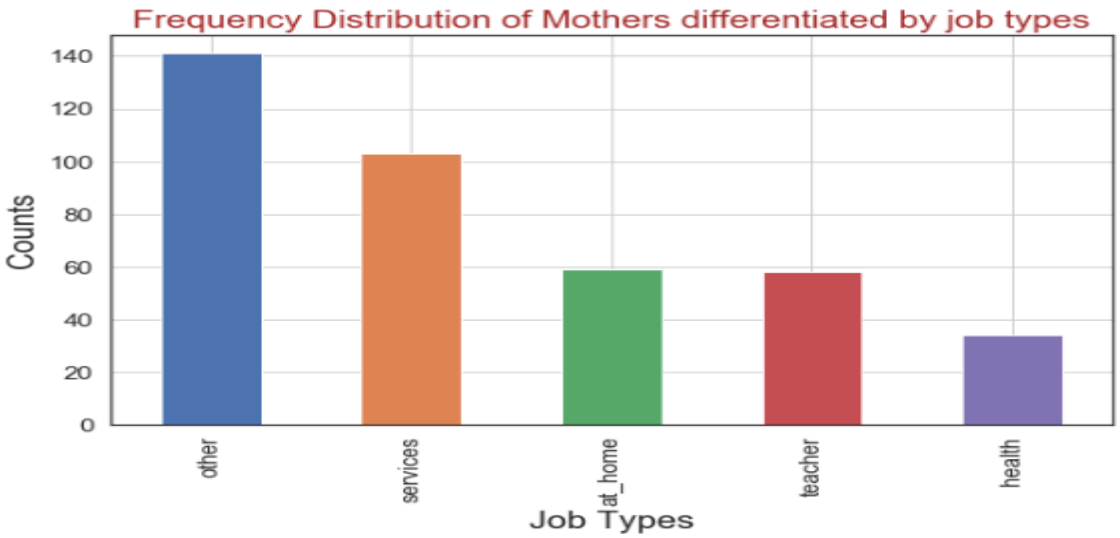

Fig II

Fig II. is a frequency distribution histogram plot which shows the count of Mothers doing different type of jobs. As we can see, mothers are mostly found working in 'services' and 'other'.

Although they are giving some clear visual insights, we were not getting a clear hypothesis related with the target feature.

- **Multivariate Analysis**

  Explored 11 pair of attributes by using count-plot, swarm-plot, kernel density graph, boxplots, scatter plot and finally a correlation matrix to find the most co-related

features in the dataset. Most of the attributes are compared with the target feature G3 to get plausible hypothesis and good insights for further analysis.

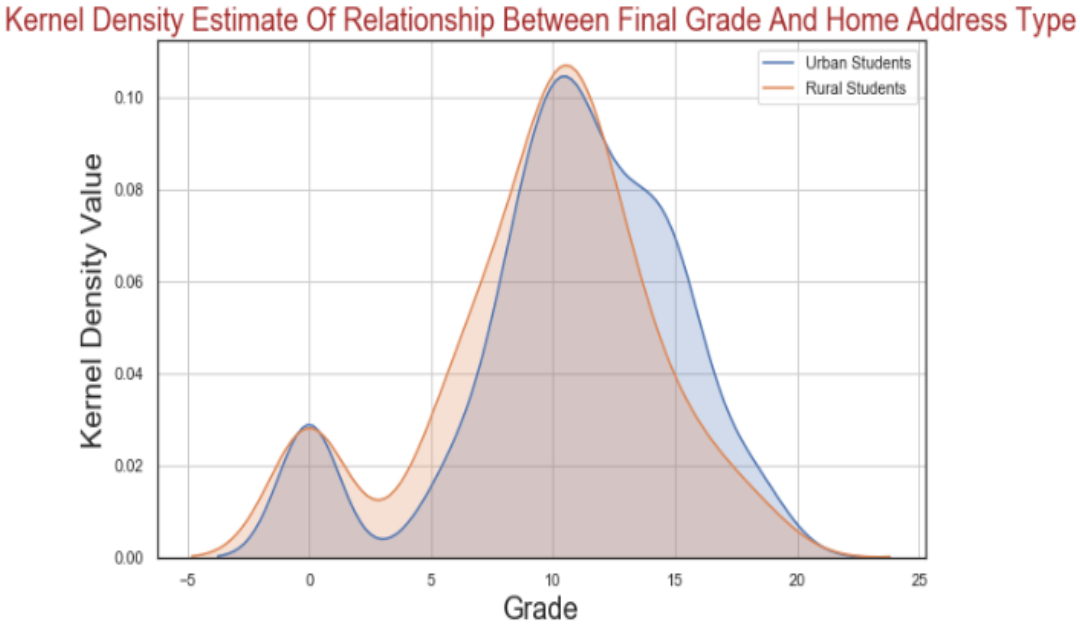

Fig III

Fig III shows Kernel Density Estimate between 'final grade' and 'home address type'.

- **Plausible Hypothesis** - Do urban students score higher than rural students?

  Fig III shows that the average marks scored are almost the same for both Urban and Rural students. Therefore, both type of students scored the same.

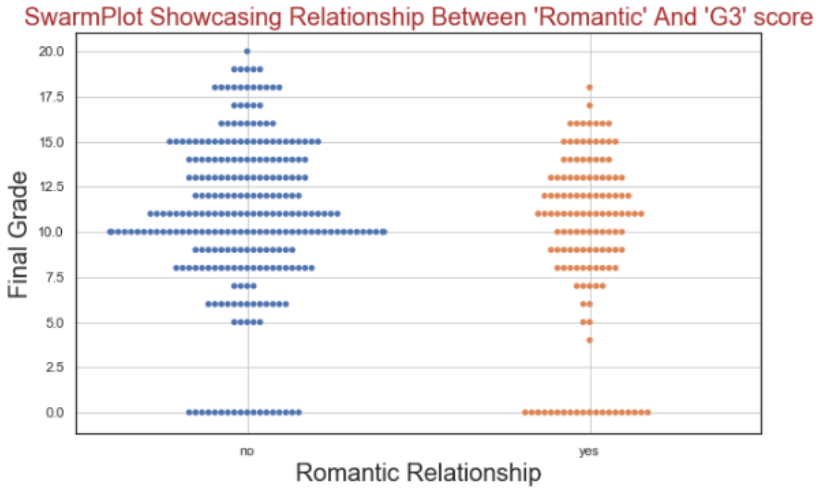

Fig IV

Fig IV shows a swarm-plot which shows relationship between 'romantic' and 'G3' attribute.

- **Plausible Hypothesis** – Student with Romantic relationship scores less.As we can see from Fig IV, students in romantic relationship scored less than those who aren't in romantic relationship.

To be able to work with both KNN and Decision tree, the *sklearn* library requires all values to be of numerical datatypes.

Firstly, we created a new column 'final grades' from 'G3' by converting numerical values into a binary factor variable with classes 'Pass' and 'Fail'. Students who scored less than 10 comes under 'FAIL' category and those who scored more than that comes under 'PASS' category.

Secondly, to change all the data values into numeric form, we mapped each and every string related attribute. Removed irrelevant attributes like 'final_grades','G1','G2', 'G3' and 'school'.

Lastly, we split the data into 'X_final' and 'Y_final' datasets. 'X_final' dataset contains all the attribute except the target attribute while 'Y_final' contains only the target attribute.

Data Modelling

After exploring the whole dataset, we decided to go with Classification method. Used both KNN Classification Model and Decision Tree Model to predict weather the student will PASS or FAIL.

From *sklearn* we imported test/train split to split both 'X_final' and 'Y_final' data into training and test set at 3 different ratios.

We split the data into -

∗ 50% for training and 50% for testing

∗60% for training and 40% for testing

∗ 80% for training and 20% for testing

- **K-Nearest Neighbour's**

This model calculates the distance of new data point to all other training data points. The distance can be of any type for e.g. Euclidean or Manhattan . It then selects the K-nearest data points, where K be any integer. Lastly, it gives the data point to the class to which the majority of the K data points belong.

Imported K NeighbourClassifier from *sklearn* to create a KNN model for each of the test train split.

We used the value of 'k' distance as our parameter for KNN model. Tuned the parameters by using '*np.arange'* function and for loop .

For each of the test/train split, we created a test and train accuracy vs n-neighbours line plot to get the best k value.

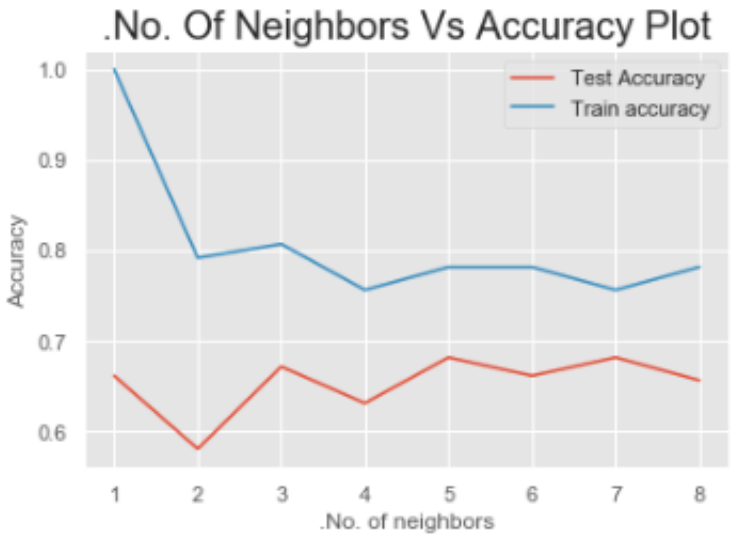

Fig V: N_neighbours vs accuracy plot for 50/50 test train split

As we can see in Fig V, the training and test accuracy is coming out to be the best at neighbours =7. The best performance for KNN Model 50/50 test train split is around 7 neighbours (k=7).

Similarly, we found the best values of 'k' for other splits.

Hyper-tuned the value of metric ='minskowski' and p = '2' along with 'k' for a better model accuracy.

After hyper-tuning, we check the performance by testing the accuracy of each KNN model at every split.

- **Decision_Tree**

Decision tree uses tree representation to solve the issue. Each node in a decision tree represents an attribute and each and every leaf node represents a class label.

For each of the test/train split, we did the same process as we did for KNN models, for tuning the parameters we used 'criterion', 'max_features' and 'max_depth'.

'Criterion' - used to measure the quality of a split. "gini" of for Gini impurity and "entropy" for the information gain.

'max_features' – selects the features for best split.

'max_depth' – gives the maximum depth of tree.

These parameter tuning helps in overfitting of model.

Used 'RandomizedSearch() function which implements a "fit" method and helps in doing cross validation of dataset. This helps in improving the overall accuracy of model.

After hyper-tuning, we check the performance by testing the accuracy of each Decision Tree model at every split.

RESULTS

Performance report for both KNN and Decision Tree model at each Test/Train split –

KNN Model Performance Report			
Parameter/Data Splitting	*50/50*	*60/40*	*80/20*
True Positive	115	93	47
False Positive	45	35	16
True Negative	20	17	10
False Negative	18	13	6
Error Classification Rate	31.80%	30.37%	27.80%
Precision	0.66	0.67	0.71
Recall	0.68	0.7	0.72
F1-Score	0.65	0.67	0.7

Decision Tree Performance Report			
Parameter/Data Splitting	*50/50*	*60/40*	*80/20*
True Positive'	133	87	53
False Positive	65	40	26
True Negative	0	19	0
False Negative	0	12	0
Error Classification Rate	32.82%	37.34%	32.90%
Precision	0.45	0.59	0.45
Recall	0.67	0.63	0.67
F1-Score	0.54	0.6	0.54

Best Model with respect to the following parameters

- **Precision -** True Positive /True positive + False Positive. Best model is KNN with (80/20) split.
- **Recall** -True Positive /True positive + False Negative. Again, KNN Model with (80/20) split performs the best.
- **F1** =2/(1/Recall) + (1/Precision). Weighted average of precision and recall.

- **Error Classification Rate** is coming out to be the lowest in KNN Model(n=5) with 80/20 test train split with 27.8 %.

DISCUSSION

	Accuracy Score	Error Classification Rate	Model Type
2	0.721519	0.278481	KNN Model 80/20 split
1	0.696203	0.303797	KNN Model 60/40 split
0	0.681818	0.318182	KNN Model 50/50 split
3	0.671717	0.328283	Decision Tree Model 50/50 split
5	0.670886	0.329114	Decision Tree Model 80/20 split
4	0.626582	0.373418	Decision Tree Model 60/40 split

Sorted the models based on Error Classification Rate. So that we would be able train the model better by putting weights on misclassified predictions.

Analysis for both KNN and DT, it is evident that due to the lack of evidence, the model performance is affected. Despite of the shortcomings, both KNN and DT had an above average performance according to Precision and Recall, but the best model performance was showcased for KNN Model 80/20 test train.

CONCLUSION

KNN model at k=5 with 80/20 test train data splitting is the best performing model. Precision, Recall and F1-score are the highest when compared with other models. Moreover, Classification Error Rate is the lowest as misclassification affects the model more as it would predict failed student to pass or vice-versa.

REFERENCES

Dataset link

https://archive.ics.uci.edu/ml/datasets/student+performance

Decision Tree Model results

```
*************Decision Tree (80/20 split)**************
Confusion Matrix:
 [[ 0 26]
 [ 0 53]]
Error Classification Rate:0.329113924051
Classification Report:
                precision    recall  f1-score   support

             0       0.00      0.00      0.00        26
             1       0.67      1.00      0.80        53

   micro avg       0.67      0.67      0.67        79
   macro avg       0.34      0.50      0.40        79
weighted avg       0.45      0.67      0.54        79
```

```
*************Decision Tree (60/40 split)**************
Confusion Matrix:
 [[12 40]
 [19 87]]
Error Classification Rate:0.373417721519
Classification Report:
                precision    recall  f1-score   support

             0       0.39      0.23      0.29        52
             1       0.69      0.82      0.75       106

   micro avg       0.63      0.63      0.63       158
   macro avg       0.54      0.53      0.52       158
weighted avg       0.59      0.63      0.60       158
```

```
*************Decision Tree (50/50 split)**************
Confusion Matrix:
 [[  0  65]
 [  0 133]]
Error Classification Rate:0.328282828283
Classification Report:
                precision    recall  f1-score   support

             0       0.00      0.00      0.00        65
             1       0.67      1.00      0.80       133

   micro avg       0.67      0.67      0.67       198
   macro avg       0.34      0.50      0.40       198
weighted avg       0.45      0.67      0.54       198
```

KNN Model results

```
*************KNN (50/50 split)**************
Confusion Matrix:
 [[ 20  45]
 [ 18 115]]
Error Classification Rate:0.318181818182
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.31      0.39        65
           1       0.72      0.86      0.78       133

   micro avg       0.68      0.68      0.68       198
   macro avg       0.62      0.59      0.59       198
weighted avg       0.66      0.68      0.65       198
```

```
*************KNN (80/20 split)**************
Confusion Matrix:
 [[10 16]
 [ 6 47]]
Error Classification Rate:0.278481012658
Classification Report:
              precision    recall  f1-score   support

          0       0.62      0.38      0.48        26
          1       0.75      0.89      0.81        53

   micro avg       0.72      0.72      0.72        79
   macro avg       0.69      0.64      0.64        79
weighted avg       0.71      0.72      0.70        79
```

```
*************KNN (60/40 split)**************
Confusion Matrix:
 [[17 35]
 [13 93]]
Error Classification Rate:0.303797468354
Classification Report:
              precision    recall  f1-score   support

          0       0.57      0.33      0.41        52
          1       0.73      0.88      0.79       106

   micro avg       0.70      0.70      0.70       158
   macro avg       0.65      0.60      0.60       158
weighted avg       0.67      0.70      0.67       158
```