IIA Research – Predict Student's Success

Damiano Pellegrini
mat. 886261
repr.: Dr. Giuseppe Vizzari
Milan, Italy
contact@damianopellegrini.dev

***Abstract*— Various Machine Learning techniques are explored and joined together to try and predict students' success.**

***Index terms*—Multi-class classification, Imbalanced classes, Academic performance, Machine Learning**

I. Introduction

In this paper I'm explaning my process to try and predict student's academic success via ML models. As for the dataset I'm gonna use the same as described and analysed in the *Predict students' dropout and academic success* paper [1] alas of much smaller size.

Since data has already been thoroughly preprocessed as stated in the paper [1] not much has to be done. Still I wanted to analyse it statistically before start feeding it to any ML model.

A. Paper overview

In Section II is detailed how I approached the problem using statistics theory and what I concluded from it, next in Section III I describe what and why certain models where chosen and how they were optimized, finally explaining my final classifier. Then in Section IV I expressed what I observed.

II. Statistical analysis

I firstly centered the data to be normally distributed, since it was required by some models and for further analysis. Next, I calculated each feature's correlation with the target. This resulted in features with an $|\varepsilon| < 0.1$, which showed that some of them contribute very little to the discrimination of students. *See [2] for a statistical visualization of each feature of the dataset*. The way forward is to ignore these features or tune hyperparameter to ignore them.

III. Classifier configuration

I used various model to benchmark and later ensemble to make a meaningful prediction, many of the following were selected among those shown in [3] and [4] from scikit-learn docs.

A. Models & Techniques explored

1) *LogisticRegression:* Despite the name is a linear classification algorithm that works with multiple classes. Useful to set a baseline and see if data can be fitted linearly. It is also quite fast.

2) *SVM - SVC:* Support Vector Machines work well with high-dimensionality problems and multi class targets, also the choice of a linear kernel derive from hyperparameter tuning explained in Section III.B.

3) *K-NearestNeighbors:* Used as a baseline in the sense that is used whether or not data is linearly separable.

4) *Decision Tree:* Divides the features' space. Max depth is limited to avoid overfitting.

5) *Gradient Boosting:* Boosting technique which uses multiple decision trees, whom the next corrects previous' tree error. Very robust against overfitting.

6) *Random forest:* Bagging technique, a decision trees ensemble robust to overfitting that works well with large datasets.

7) *Adaboost:* Boosting technique, iteratively adjust errors, giving an increasing weight each error. Useful with complex data.

8) *Perceptron:* Not very useful since we have more than 2 classes, uses L2 penalty.

9) *Multi-layer Neural Net:* Very efficient in capturing complex patterns in a dataset but may be subject to overfitting.

B. Hyperparameter Tuning

After scoring every single model with default parameters I decided to tune its hyperparameters to achieve a higher accuracy. To find optimal values I used a `5-fold GridSearch` [5] which scored the models for various combinations.

C. Ensemble using Voting

Finally I wanted to try improving on accuracy by ensembling a more sofisticated model using a "soft" `VotingClassifier`. First, I kept only models with an accuracy over the threshold (accuracy $> \sim 75\%$). Overall accuracy didn't seem to improve that much.

IV. Conclusion

In conclusion, I got a very promising number of 78% accuracy in classifying the dataset. Further dissecting the metrics as seen in Figure 1 unfortunately gives a different picture. The model perform relatively well in identifying actual graduates, having both high precision and recall, the same could be said for the dropouts. When it comes down to the enrolled class however the model performs way worse. This suggests that the model may need further improvement, needs a more balanced distribution or needs more features that discriminates better the population.

```
Classifier Name             Accuracy
LogisticRegressionCV          74.66%
SVC                           73.30%
KNeighborsClassifier          71.18%
DecisionTreeClassifier        69.39%
GradientBoostingClassifier    78.03%
RandomForestClassifier        77.67%
AdaBoostClassifier            74.86%
Perceptron                    65.21%
MLPClassifier                 75.63%
Classifier over thresh (> 0.75): dict_keys(['xgb', 'rf', 'mlp'])
-----------------------------------
VotingClassifier              78.50%
-----------------------------------
Classification Report:
              precision    recall  f1-score   support

     Dropout       0.79      0.73      0.76       284
    Enrolled       0.62      0.37      0.46       169
    Graduate       0.78      0.93      0.85       432

    accuracy                           0.76       885
   macro avg       0.73      0.68      0.69       885
weighted avg       0.75      0.76      0.75       885
```

Figure 1: The final classifier metrics

References

[1] V. M. M. Realinho Valentim and L. Baptista, “Predict students' dropout and academic success”. 2021.

[2] “Statistical visualization”. 2021.

[3] “Classifiers comparison”.

[4] “Supervised Model booklet”.

[5] “Demonstration of multi-metric evaluation on cross_val_score and GridSearchCV”.