Supervised Learning on Grades Prediction

Phornphawit Manasut

Abo Akademi University
e-mail: phornphawit.manasut@abo.fi

Abstract. The grades of students and their progression in class are of concern to various parties including lecturers, administrators, parents and students themselves. This report attempts to predict student final grades while class is still going on and also try to demonstrate the curse of dimensionality in supervised learning. The author also attempts to show downside of analysis that includes data that should not be used.

1. Introduction

Grade is used as a mean to measure student's performance and understanding of class that he or she took. It can also be used as a tool to measure lecturer's ability to teach his or her students. Therefore, it is often times in the best interest of both parties that student performs well in the class. Failure does not benefit either party. However, time is a limited resource and lecturers must prioritise their time to few truly struggling students. This is difficult to do based on looking at the data alone. Supervised Learning methods can, therefore, be used to predict students which may fall in the unsatisfactory grades and prioritise helping them. This report uses the data from Moodle Site and try to create the best possible set of features to predict grades. This report aims to answer the following question: how bad is the curse of dimensionality?, can we accurately predict final grades based on data just after week 5?, how does the existence of non-relevant data affect the evaluation of results?

```
new_score = df[q_cols + mp_cols + pr_cols].sum(axis=1)
print('All same data:', pd.Series(np.isclose(new_score, df['Week8_Total'])).unique().shape[0] == 1)

All same data: True
```

Fig. 1. Code and verification for summing of grades.

2. Data Observation

Our original data set from moodle has 48 columns and can be categorised into 4 groups: ID, Grades, Site Statistics, Grade Label. ID will not be used in this analysis.
Following characteristics is observed:

1. Summing of all grades from before week8 will give values of Week8_Total column. This total corresponds directly to the Grade label. The verification of the sum used default numpy.close function in order to bypass floating point error. This is shown in Fig 1 and Fig 2.

2. Week1_Stat1 column only has 0 as its value, therefore, has no variance to the outcome Grade. Therefore the column shall not be used in the analysis

3. There are 2 groups of 0: one who truly try and fail, one who just want to see the

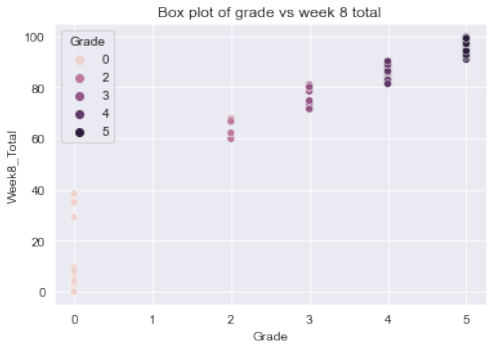

Fig. 2. Week 8 show how summing of grades can be used for rule-based check.

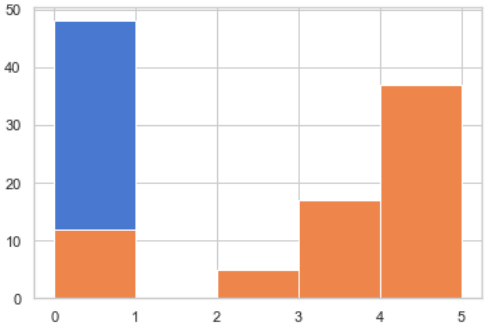

Fig. 3. Grade distribution before and after the discard. Blue section is the discarded data points.

course content on Moodle. The inclusion of both groups cause extremely misleading analysis. This shall be demonstrated in the Result and Evaluation section.

3. Data Processing

In this section we shall go through what was done to the data before the prediction phase.

Previously, we mentioned the existence of 2 groups of students with Grade 0. The author detect those who applied to the course just to see the content by using all grades columns, if all these grades are 0 then the author assume that they are just here to see the content and therefore, did not attempt to earn any grade. We shall call the dataset with all rows ”df”. We shall call the dataset with discarded rows as ”df2”. The distribution of grades before and after is shown in Fig 3. ”df” has total number of rows of 107 while ”df2” has 71.

In data processing, we also create 18 new columns. They can be put into following categories: Weekly Statistics, Set Grades, Week 5 Grade Cumulative, Week 3 Statistics Cumulative, Week 5 Statistics Cumulative. The reason for Week3 and Week5 cumulative groups is because this is the period where first set of mini project, peer review, and quiz is done. These columns are created using the sum of grades or statistics. There were no multiplication involved due to lack of time. Set Grades columns are created by summing same set of mini project, peer review and quiz together: example is first mini project, first quiz, and first peer review grades are summed up.

4. Supervised Learning

Supervised learning is a backbone of machine learning. There are various methods but the ones selected were RandomForest and SVM.

For SVM we use linear kernel in order to try and capture linear relationship which was seen in Fig 2. The relationship among behind variables may not be linear but since the final grade simply involve summing up all the grades, this may be the best one available with the current data processing. Had the author been more adventurous, perhaps polynomial features might be interesting.

For RandomForest, depending on the number of columns used, we mainly limit tree depth to maximum of 5 and maximum features used split node to be from 3 to 10 depending on, once again, the number of columns.

There are 3 factors influencing the training and testing of Supervised Models: Dataset, Is Predicting Future Grade, Columns Used.

Dataset factor is straight forward. Did we use ”df” or ”df2”?

Is Predicting Future Grade is different. It asked the question: Are we trying to train the model with data from the entire timeline or trying to predict future grade with data up to week 5? The former question is trying to ascertain how well the models capture the underlying heuristic with more and more dimensions being used.

Columns used is simply the set of columns and features used. We have 3 sets of columns for when Is Predicting Future Grade = False: Week8 only, All Grades no sum, Set Grades. We have 5 sets of columns for when Is Predicting Future Grade = True: Set 1 and 2 Grade Total, Raw Grades from week 1 to 5, Raw Grades and Stats from week 1 to 5, Grade set 1 and 2 and Weekly Sum Stat from week 1 to 5.

These 3 factors shall be used to name the training testing type. Example is "Dataset_IsPredictFuture_Columns"

We then ensure the reliability of models and accuracy by using K Fold Validation with k=10.

5. Result and Evaluation

Table 1 shows the mean accuracy from training and testing data set for types of prediction, dataset, and columns used.

We pick 1 model and training instance to analyse for each type of IsPredictFuture. Fig 4, 5 and 6 represent figures related for when IsPredictFuture=False. Fig 7, 8, and 9 represent figures related for when IsPredictFuture=True.

6. Discussion

Let us take a look at Table 1 and start discussion on why keeping the separate set of grade 0 can make analysis misleading. If we take a look at all the accuracies for models trained on "df" vs "df2", we can see that the number is lower comparatively on all instance of training and testing. This can make analysis misleading because we are predicting students' grades who do not need our attention and our accuracy is actually higher than it's supposed to be, making us think that the model can predict students' grades well enough for group of students we care about. Therefore, this subset of 0 Grades should definitely be excluded.

Let us then take a look at Table 1 once again to talk about curse of dimensionality. We demonstrated here that we a single column, we can make the model infer on the human heuristics grading rule but it is still limited to the

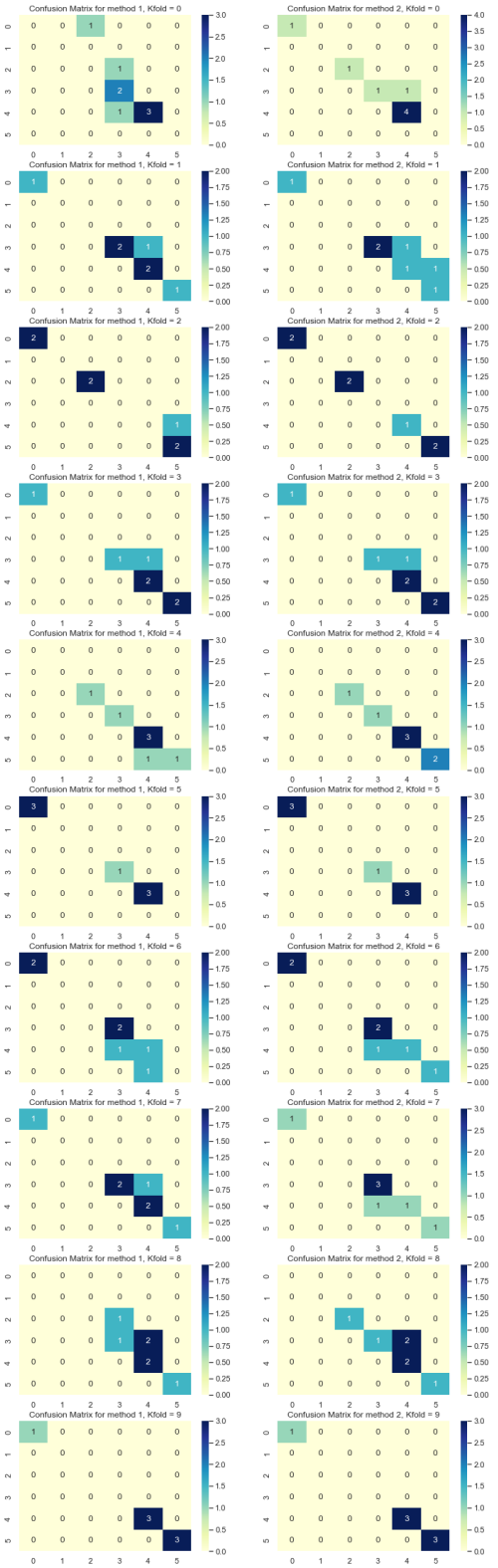

Fig. 4. Confusion matrix for all k fold for df2_false_allgradeset model

Table 1. Result

Name	RF Mean Accuracy (%)	SVM Mean Accuracy (%)
(Dataset_IsPredictFuture_Cols)	(Train, Validation)	(Train, Validation)
df_false_week8only	(100.0, 99.0)	(93.04, 92.55)
df2_false_week8only	(100.0, 98.57)	(96.40, 97.14)
df_false_allgraderaw	(100.0, 80.64)	(93.66, 91.73)
df2_false_allgraderaw	(100.0, 73.03)	(90.61, 87.32)
df_false_allgradeset	(97.93, 84.10)	(93.66, 93.45)
df2_false_allgradeset	(97.81, 81.96)	(90.76, 88.75)
df_true_gradeset	(82.97, 74.73)	(67.29, 66.45)
df2_true_gradeset	(73.23, 63.39)	(55.87, 53.92)
df_true_gradesraw	(84.11, 76.8)	(79.03, 75.90
df2_true_gradesraw	(76.68, 61.96)	(72.13, 61.78
df_true_gradesstatsraw	(95.01, 69.45)	(94.49, 58.18
df2_true_gradesstatsraw	(92.80, 53.82)	(93.26, 40.89
df_true_gradessetstatsweekly	(85.35, 72.81)	(79.44, 71.09
df2_true_gradessetstatsweekly	(78.24, 63.39)	(72.61, 56.61)

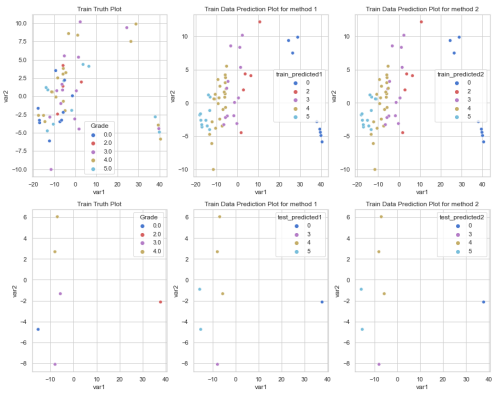

Fig. 5. Train-Test PCA-Label Plots for model index 3 in kfold for df2_false_allgradeset

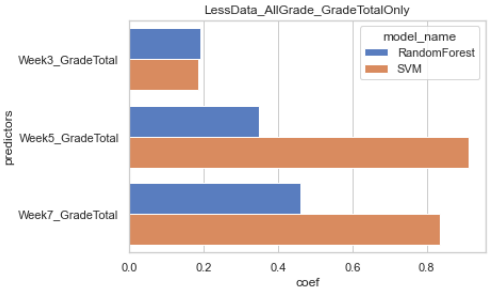

Fig. 6. Feature Importance Plot for models index 3 in kfold for df2_false_allgradeset

data it is trained on. However, when we use all grades which consist of 9 columns, the model accuracy dropped drastically. This is also supported by summing grade set which consist of 3 columns, the model accuracy drop quite drastically but not as much as 9 columns. SVM, however, seems to do particularly well against higher dimension but it still loses accuracy, just not as much as Random Forest.

Lastly, we can see from Table 1 that some of the training instances is wildly overfitting. The author tried to limit several parameters for training model but given the limitation of data size, there is only so much that can be done. Maybe the author could try generating synthetic data but that is out of scope of this mini project and the author does not have enough time.

For grade rule learning, we would like to know more detail about how the model use features given for "df2_false_allgradeset" instance Therefore, we put in additional plots. From the Fig 6, it is interesting to see how SVM place Week5 Grades (all data from 2nd set of Quiz, Mini Project, and Peer Review) ahead of

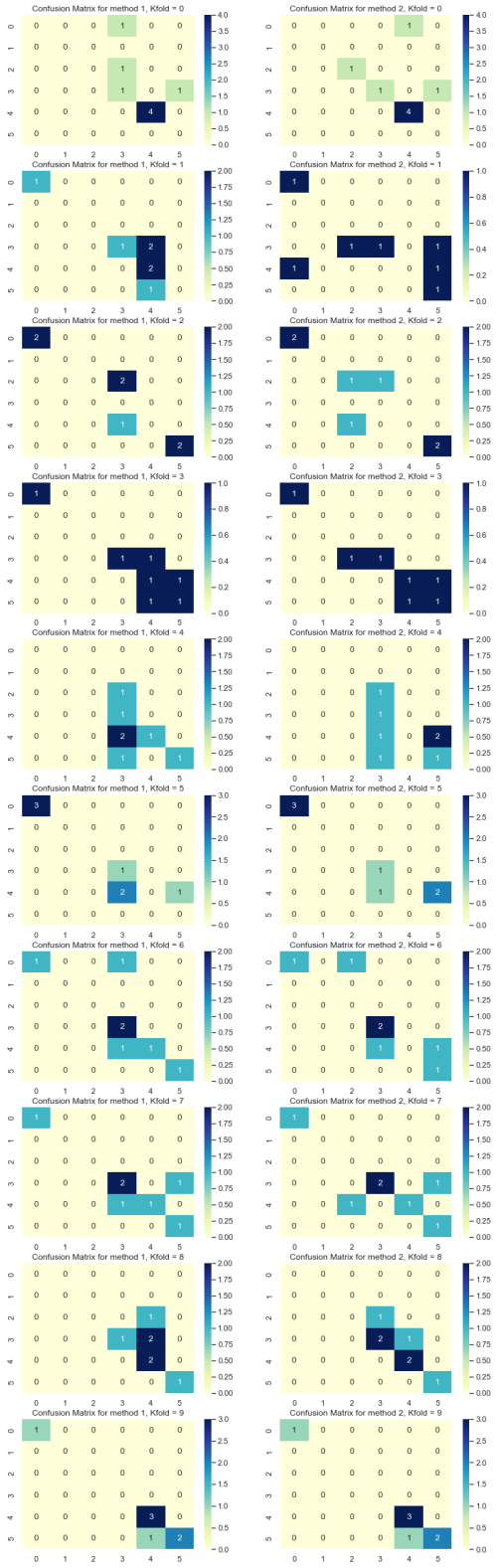

Fig. 7. Confusion matrix for all k fold for df2_true_gradesraw model

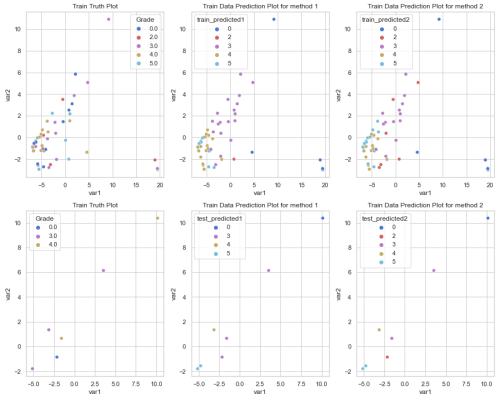

Fig. 8. Train-Test PCA-Label Plots for model index 7 in kfold for df2_true_gradesraw

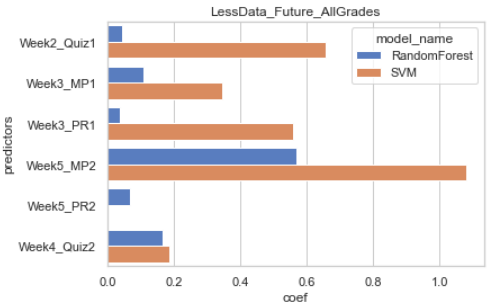

Fig. 9. Feature Importance Plot for models index 7 in kfold for df2_true_gradesraw

Week7 but RandomForest places Week 7 first as the most important. In Fig 5 where we perform PCA on the data to reduce its dimension to 2 dimension and plot to see grades distribution and how our train test models predict compared to this distribution. As we can see, our model try to be very linear in its decision boundary (middle and right plots) whereas the actual data relationships is much more complex than our initial assumption (left plot). Perhaps more time should have been put to investigate the relationship between predictors and the label.

For future grade prediction, we notice that for "df2", the "df2_true_gradesraw" training instance obtain a not so bad accuracy comparatively to its peers even if it is still terrible. Therefore, we decided to do further analysis of

the instance. In Fig 9, we can see that SVM and RandomForest is quite similar in its prioritisation of features, with Week5_MP2 being the most important feature. However, they differ in that 2nd most important for SVM is Week3_MP1 whereas it's Week4_Quiz2 for RF. The time factor might be more important for RandomForest to infer grade compared to SVM.

7. Conclusions

We have answered all the answers proposed in the introduction. We demonstrated the curse of dimensionality by trying to let the models learn of human grade rules from various set of features. We have shown how accurately we can predict the finals grade of students using only grades up to 5th week of study. We have also explain why the usage unrelated data is terrible for the analysis.

One very interesting is how RandomForest tends to place more importance in a feature that occurs later in time than SVM, even if it's by a little margin of 1 or 2 place.

One technical pitfall that the author fell into is assuming that relationship of predictors to label may be linear which may not be true given the result in the PCA plots.

Further work could be done like trying to generate synthetic data to make the model not overfit or perform better hyperparameter tuning.