Machine Learning Final Report

R26121015 RUI-TING,HUNG

January 10, 2025

Motivation

From the competition in HW2, the training data includes Date, time, weight, feature_0 to feature_78, and responder_0 to responder_8. I used all these features (except responder_6) to train the model. Afterward, when inspecting feature importance, I found that responder_0 to responder_8 had a significant impact on predicting responder_6.

So, I believe that if the "predicted values" can be used as features for training, it might yield good results.

Methodology

I will implement the following method for both prediction and classification problems:

- **For prediction problems:** I will first obtain the predicted values, denoted as $\hat{y}$, and add them to the original features before performing another prediction.
- **For classification problems:** I will first obtain the probabilities for each class, denoted as $\hat{p}$, and add them to the original features before performing another classification.

Real Data Analysis: Prediction

I used the built-in Python wine dataset for this analysis. I chose "proline" as the response variable, while all other features were used for training. Two models were applied: linear regression and random forest.

First, I used these two models to make predictions, generating predicted values denoted as $\hat{y}_{\text{lm}}$ (from linear regression) and $\hat{y}_{\text{rf}}$ (from random forest). Next, I added $\hat{y}_{\text{lm}}$ and $\hat{y}_{\text{rf}}$ to the original features and performed another round of predictions. Table 1 and Table 2 summarize the results.

Table 1: Linear Regression

	MSE(original)	MSE(add features)	R square(original)	R square(add features)
training set	33605.46	1963.63	0.629	0.978
testing set	42923.17	25739.82	0.948	0.993

Table 2: Random Forest

	MSE(original)	MSE(add features)	R square(original)	R square(add features)
training set	4671.38	556.34	0.948	0.993
testing set	29218.06	27689.96	0.769	0.781

Table 1 and Table 2 show the results, where the values highlighted in red represent the training results using predicted values as additional features. From the tables, it can be observed that using predicted values as features improves performance in both MSE and R square. This indicates that incorporating predicted values as features indeed enhances prediction performance on this wine dataset.

Real Data Analysis: Classification

I also used the built-in Python wine dataset for this analysis. I chose "class" as the response variable, while all other features were used for training. Three models were applied: logistic regression, SVM and random forest.
Similar to the steps I took for prediction, I first perform classification with these three models, and return the model's predicted probabilities for classes 0 to 2. These probabilities are denoted as $\hat{p}_{0\text{lr}}$, $\hat{p}_{1\text{lr}}$, $\hat{p}_{2\text{lr}}$, $\hat{p}_{0\text{svm}}$, $\hat{p}_{1\text{svm}}$, $\hat{p}_{2\text{svm}}$, $\hat{p}_{0\text{rf}}$, $\hat{p}_{1\text{rf}}$, and $\hat{p}_{2\text{rf}}$. I then add these 9 probability features to the original features and perform another round of classification. Table 3 will display the performance of each model during the first classification.

Table 3: classification accuracy

	Accuracy
logistic regression	0.96
SVM	0.69
random forest	0.98

From Table 3, it can be observed that both logistic regression and random forest already performed well on the test set. Therefore, I focused on the SVM model and added the 9 predicted probability features mentioned earlier to the training. The classification accuracy for this iteration was 0.7, with no significant improvement.

I believe this lack of improvement might be due to the addition of too many features, which made the training more difficult. As a result, I selected the top 10 most important features, 6 of which came from the 9 newly added probability features. After training the SVM model again with these 10 features, the classification accuracy improved to 0.97.

Conclusion

For classification and prediction tasks, I found that adding "prediction" or "classification" features to the wine dataset indeed improved the performance of both prediction and classification. As for clustering problems, although I didn't implement it, adding pre-clustered features and then retraining the model should also enhance the clustering effectiveness.

This improvement might be due to the relatively simple and straightforward nature of the wine dataset. If this method were applied to more complex datasets, the results might not be as significant as what I demonstrated. However, if you ever encounter a dataset where you have no clue about adding features, you can try this approach—it might yield unexpected results.