CSCI-P556
Fall 2018
Assignment 3
Due 11:59PM, Nov. 2, 2018

Archish Ramesh Babu (arameshb)

November 5, 2018

1 Introduction

In this assignment we are being given the following four data files:

1. a3-train.data
2. a3-train.labels
3. a3-test.data
4. a3-test.labels

There are 2,000 training rows and 600 test rows, each row has 500 features, and there are only two labels, -1 and 1.

2 Exploratory Data Analysis

1)Missing Value: There were no missing values observed in both the test and train datasets.

2)Balanced Labels: If the dataset is unbalanced then the majority of the popular classification algorithms like Logistic regression, Random Forest etc. will skew towards the class that has majority of the observation and we would have to treat it with techniques like Undersampling or Oversampling to name a few. Both our test and train dataset are perfectly balanced with each label accounting for 50 Percentage of the data.

3)Univariate Analysis : Upon looking at the distribution of each column we can see that there are no obvious outliers in the data.

4) Unique values in column : Since the columns are encoded we need to check if some of the variables could be categorical. The minimum number of distinct values present in a column is 5, it is hard to conclude if the variable is categorical based on this result, so I have decided to go ahead with considering them as numeric.

One major observation is that majority of the data is not very spread.

3 Baseline Models

Having identified a suitable feature set to train our models with, you will train at least three baseline models with the default parameters to get an idea of what is the minimum performance that can be

achieved before performing feature engineering and optimizing the model's hyperparameters. I have selected 4 models for calculating the Baseline, the four are

1: Logistic Regression: Logistic regression is a very good model to baseline results because it does not take a lot of time to run compared to other models in its category, also, this would be a good model to improve because there is a lot of variables which would result in overfitting and we can use L1 norm plus regularization to account for this
Accuracy of Train Data: 0.741
Accuracy of Test Data: 0.5866
Conclusion: The model is overfitting the data hence there is a need to perform regularisation.

2. Random Forest: Random Forest would be a good model because it not only works well when there are a lot of features but also works provides variable importance which is very useful during feature engineering.
Accuracy of Train Data : 0.986
Accuracy of Test Data : 0.62
Random Forest is performing better than logistic regression as expected given the number of features in the data. But there is a lot of overfitting taking place which could be observed from the difference inaccuracies between the test and train. Grid search would be very useful method to resolve this issue.

3. KNN: Based on the EDA we observed that for majority of the features the values were close to one another and KNN would be very good model to fit this.
Accuracy of Train Data : 0.8265
Accuracy of Test Data : 0.69666
The KNN is performing a lot better than the random forest, which is a bit surprising and the data is also not that overfit. Again Grid search and feature engineering should improve this result.

4. Gradient Boosting: Gradient Boosting gives the best accuracy among the baseline models, this is because of its ability to deal with overfitting
Accuracy of Train Data: 0.962
Accuracy of Test Data: 0.7466
Post feature engineering the accuracy of Gradient Boosting is expected to the highest.

4 Feature Engineering

Based on the results from the Baseline models we can observe that there is a lot of overfitting happening, hence it is very important to control with that first.

The best way to perform feature engineering is to understand the importance of columns based on Business sense and along with the accuracy from the models using those columns, since we don't have business understanding of the variables in this case, I have decided to use an automated method to identify the best list of features using RFECV.

This method repeatedly builds multiple models and identifies the best set of features based on Cross Validation accuracy. It identifies important features during each iteration and keep them aside and then adds and removes features iteratively till all the features have been exhausted. The RFECV is an improvement over the RFE method because it all provides the optimal number of features by Cross-validation.

Important features : 28, 48, 64, 105, 128, 153, 241, 281, 318, 336, 338, 378, 433, 442,451, 453, 472, 475, 493

The model used for RFECV was the random forest classifier because it does not only look at the standalone performance of a variable but also looks at how it performs when combined with other variables.

5 Model Building

In this section you will build at least three models in which you will try to achieve the highest possible performance on the test set. **Your models performance on the test set will be taken into consideration when we grade this assignment**. You are free to use whichever models and techniques (stacking, ensembles, etc.) you want and are not restricted to the ones that we have covered in class.

1: Logistic Regression: We already know that there was a lot of overfitting taking place and we have accounted for some it using Feature Engineering. To get the best model for Logistic regression, we are performing Grid search by varying Lambda value and the type of penalty.
Best parameters after Grid Search are C: 0.0009, penalty: l1
Accuracy on Train Data for Logistic Regression after Grid Search = 0.6165
Accuracy on Test Data for Logistic Regression after Grid Search = 0.5916
Conclusion: The results are no longer overfitting but the accuracy has not improved.

2. Random Forest: We have already done Feature Importance using random Forest so in the hyperparameters we can tune the criteria and estimators. Due to parallel processing, the model runs quickly, even if it is building a lot of trees.
Best parameters after Grid Search are criterion : entropy , max_features: auto, n_estimators: 700
Accuracy on Train Data for Random Forest after Grid Search = 1.0
Accuracy on Test Data for Random Forest after Grid Search = 0.895
This is a significant improvement in the results of the Random Forest model post feature engineering and grid search, This confirms the fact that the features selected is able to fit the data well.

3. KNN: For KNN the number of Neighbours, leaf size and the weight measurement were modified
Best Hyper Parameters are leaf_size: 1, n_neighbors: 4, weights: distance
Accuracy on Train Data for KNN after Grid Search = 1.0
Accuracy on Test Data for KNN after Grid Search = 0.8916
The performance of the tuned KNN is very close to the random forest, this probably caused by the fact that the majority of the features have very close values.

4. Gradient Boosting: The results from the Gradient Boosting has a lower than the accuracy as compared to KNN and RandomForest. Overfitting is still taking place but reducing the max_depth is not taking care of this issue.
Best Hyper Parameters: 'learning_rate': 0.25, 'max_depth': 8, 'n_estimators': 100
Accuracy on Train Data for GBM after Grid Search = 1.0
Accuracy on Test Data for GBM after Grid Search = 0.8766
Lets next try to stack these results to check for improvements.

6 Stacking

Stacking is an important concept where we use results from multiple models to create a stronger model, the idea is each model will capture certain information from the data and when they join together, the combined(Stacked) model will be able to capture the entire information which individual models fail to do.

For effective stacking the results from different models must not be very correlated, from our models we observe very high correlation from the results of Random Forest, KNN and Gradient Boosting but luckily all three of them have very less correlation with Logistic Regression.

Upon running xgboost+Grid search for Stacking, we don't see any increase in the accuracy because the variables were correlated a lot.

7 Discussion

1. The data was balanced hence we could go ahead with Accuracy as the key metric
2. The values for majority of the variables were close to each other and there were no outliers or missing data
3. Baseline models showed a lot of overfitting but this was expected due to the huge feature count
4. Recursive Feature elimination helped to automatically identify the were significantly contributing to the model
5. Grid Search was very useful in tuning the parameters of all the models and the maximum accuracy was for Random Forest Model with an accuracy of 89.5 Percentage
6. Stacking is very useful where different models capture different information but in our case, since the results from majority of the models were correlated, stacking did not prove useful for our case