CS 559: Machine Learning: Fundamentals and Applications
Fall 2024 Project: Company Bankruptcy Prediction

1 Objective

The performance of logistic regression applied to the **original train data** is presented in Table 1. While the prediction of the target is highly accurate, identifying the company with $(y = 1)$ is very poor due to the imbalance (see Figure 1) of data.

data set	FF(FT)	TT(TF)	accuracy score
train	5570(39)	5(193)	0.025
test	983(7)	0(22)	0

Table 1: Logistic Regression Result Table without any modification. The second and third columns present the number of companies' true status (y) and predicted status $(\hat{h})$ as FF - $(y = 0, \hat{h} = 0)$, FT - $(y = 0, \hat{h} = 1)$, TT - $(y = 1, \hat{h} = 1)$, and TF - $(y = 1, \hat{h} = 0)$. The accuracy score in the fourth column is evaluated using Equation (1).

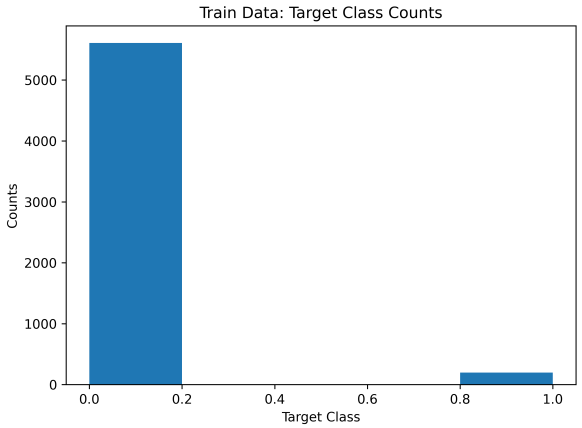

Figure 1: Train Data Target Distribution

The modeling objective is to identify the companies that will file for bankruptcy. Hereafter, the **accuracy score** (acc) is defined as how successfully the model identifies the company that files for bankruptcy as follows

$$acc = \frac{TT}{TF + TT}. \tag{1}$$

1.1 Class Competition

This project is a class competition, and teams will be ranked using the following metrics:

$$Rank = 0.3(acc_{train}) + 0.4(acc_{test}) + 0.3\left(\frac{50 - N_{features}}{50}\right)$$

where

- acc_{train} is the accuracy score of the train model from Section 3.3.
- acc_{test} is the accuracy score of generalization obtained from Section 4.

- $N_{features}$ is the number of features for the best model in Section 3.1.

The rank will be converted to the top percentile in the competition and will be used for the individual's project grade (see Section 5.3).

2 Data Description

The train and test data sets have 5807 rows and 1012 rows, respectively. The total number of given features is 95. The project aims to train a model that identifies whether a company will file for bankruptcy (the target column is 'Bankrupt?') or not. The test data set does not have the true label. Each team will submit the predicted classes for 1012 companies to the submission file.

3 Model

Since the data is insignificantly imbalanced, logistic regression could not capture the crucial features for companies with $y = 1$. Therefore, the model needs to be trained differently to account for crucial features among companies with $y = 1$. The new proposed method is to create a classification model for companies in similar conditions.

3.1 Training Data Preparation

95 features are considered too many features. The number of features **must be reduced** to increase the model's efficiency. Each team will produce a new training data set with features between 25 and 50. There are unlimited ways and numbers to extract, select, drop, and engineer features.

1. The new train data must not be multicollinear.
2. Each feature must be in its own Gaussian.

3.2 Company Characterization

Considering the data size, understanding every company's situation is quite difficult at a given time. However, knowing the general situation will greatly help model training. The easiest approach is finding the common characteristics between similar companies, and similar companies can easily be grouped by using unsupervised learning clustering techniques.

1. Cluster the training data from 3.1 into k-many subgroups where k can be up to twice the team size but not less than the team size. For example, if a team has three members, the cluster groups can be between 3 and 6. The target must not be included.
2. Report the number of companies and the balance of bankrupted companies in each subgroup.
3. Identify unique or helpful characteristics in each subgroup. Use visualization techniques to present identified characteristics.
4. Keep the cluster IDs. It will be used later in Section 3.3.

3.3 Bulding Training Models

It can be broken into two steps. The first step is to predict the group (the cluster-ID from Section 3.2) the company will likely belong to, and the second step is to classify whether the company will file for bankruptcy in each subgroup. Figure 2 displays the roadmap of the model.

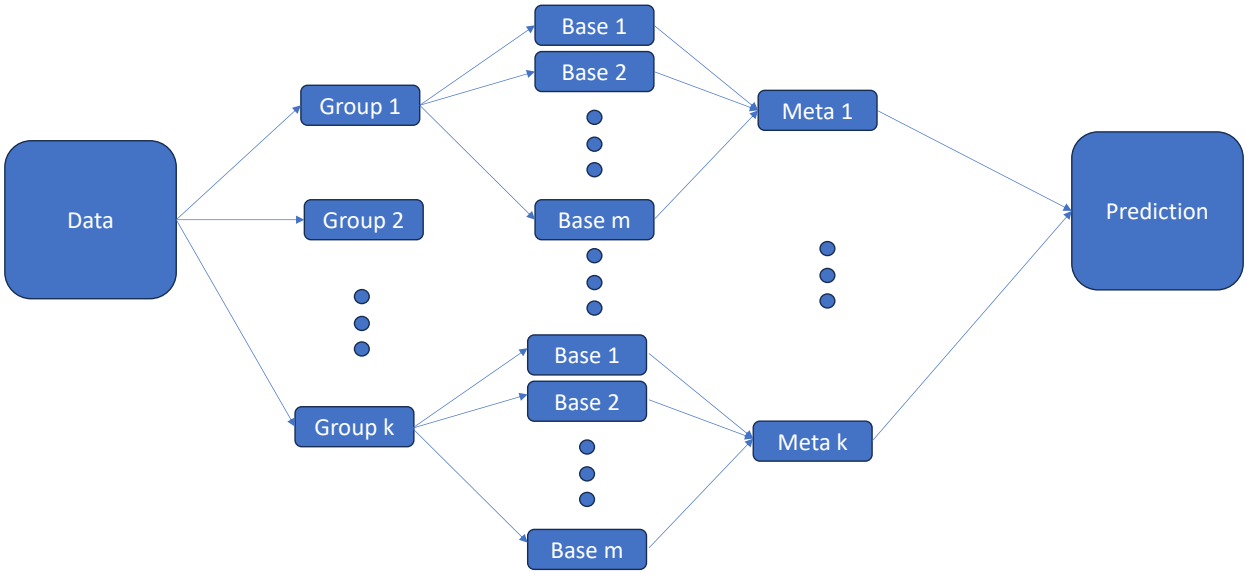

Figure 2: Stacking Method Training Model Diagram. The diagram shows the k-many subgroups, m-many base models in each subgroup, and k-many meta-models where each meta-model is a combined model of m-base models in each subgroup.

1. Build a classification model that predicts a subgroup a company will likely belong to using any supervised learning algorithm. The prediction's accuracy should be high, and it is okay to overfit. Identify the features that play important roles in this prediction.

2. Build a stacking model (with cross-validation) that predicts whether a company in the subgroup will file for bankruptcy.

    a. The base models must be non-parametric. Have three or more.

    b. Each subgroup must use the same features for all base models. However, features can be different between subgroups.

    c. Each member should build a model for at least one subgroup. The member's name, the subgroup(s) worked on, and *acc* must be reported. Provide the confusion matrix. See Table 3 for the reporting format.

4 Generalization

Transform the test data set as the training set in Section 3.1. Then, predict each company's bankruptcy status in the test data. Paste the result to the submission file (Table 2). The instructor will evaluate the accuracy score using the predicted class the team submitted.

Index	Bankrupt?
1	0
2	1
⋮	⋮
1012	0

Table 2: Submission File Example

5 Timeline, Submission, Grade Scheme

The submission of the project is due on December 10 by 11:59 PM.

5.1 Timeline

The ideal timeline of the project is broken down as follows.

1. Train Model: The train data set will be released on Tuesday, 11/12, so the training data preparation and building of a training model can be started.
2. Generalization: **The test data will be released on the second week of Project Weeks (11/26)**. The generalization should be one-time work and should not take a long time.
3. The video presentation and files must be submitted by 12/10.

5.2 Submission

Here are the requirements for submission. Each team will submit a single compressed **zip** file. The submission will require the following files inside the zip file. Name the file as *GroupNumber_CourseSection.zip*.

1. Sections 3.1 and 3.2: A single team notebook file: Clean and summarize the work. Use Markdown for comments and explanations. Name the file as *GroupNumber_TrainingData*.
2. Section 3.3: A single notebook file for individual team members. Show the work and summarize the work. **Individual files must be included in the compressed file**. If any team member fails to submit, the member will be excluded from the grading. Name the file as *MemberName_Subgroup#*. If any member worked on multiple subgroups, submit the file for each.
3. Section 4: Submit a single team's notebook file with code and the submission file as shown in Table 2. Name the file as *GroupNumber_Generalization*.
4. Result Summarization: Summarize the results as a team in the docx file.
   - Section 3.2: Report the number of subgroups, the number of $y = 0$ and $= 1$, and summarize characteristics, distributions, or properties of companies in each subgroup.
   - Section 3.3: Construct a table as shown in Table 3. In addition to that, construct a confusion table for base models and the meta-model for each subgroup.
   - Video Presentation: Record the video presentation to describe the workflow, models, and results. You can either include the video file in the zip file or provide the link in the summarization document.

   ** All reported results must be consistent with notebook files. Make sure to use **random.seed()** or **random_state=** whenever needed so the same results can be reproduced. Each notebook file must display the result. Any non-consistent results will be marked 0.

   - Name the file as *GroupNumber_CourseSection_Results*

Subgroup ID	Name of Student	Average accuracy score base models [TT(TF)]	accuracy score Meta model [TT(TF)]	$N_{features}$
1	John	0.93 [184(14)]	0.84 [167(31)]	34
2	Kim	0.78 [$\cdots$]	0.67 [$\cdots$]	49
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	Name	0.xx [$\cdots$]	0.xx [$\cdots$]	xx
team		0.xx [$\cdots$]	0.xx [$\cdots$]	xx

Table 3: Result Table Example: Column 1: The subgroup ID from Section 3.2, column 2: the name of the member who worked on the subgroup, column 3: the average of the training accuracy score (Equation 1) from beta-models in the subgroup, column 4: the training accuracy score from the meta-model of the subgroup, column 5: the number of features the member used in Section 3.3. As a team score, average *acc* scores, sum TT(TF), and average $N_{features}$.

5.3 Grade Scheme

This is how the project will be graded. Although this is a team project, each member will be graded separately, as shown below

$$Score = 0.2(Rank) + 0.4(acc_{Table3}) + 0.4\left(\frac{50 - N_{features,Table3}}{50}\right) \tag{2}$$

where $Rank$ is the top percentile of the competition, acc_{Table3} is the accuracy score of the individual's train model in Section 3.3*, and $N_{features,Table3}$ is the number of features the individual used in the training model in Section 3.3. The accuracy scores and number of features reported in Table 3 will be used. The accuracy metrics in Equation 1 will be used for the individual's grade.

Note*: If a member trained the staking model for more than one subgroup, the best-performed model in terms of accuracy will be used in the grading metric.

Note: The group performance ($Rank$) will weigh 20% of the project grade, and the remaining 80% of the grade will be evaluated from the individual's performance.

Note: This is a team project. If any member of a team is not responsive and does not return emails, texts, or calls, it is the team's responsibility to contact me. If this member was reported twice or more, the student will be out of the project and receive a zero. Please be responsive to team members.