Machine Learning Research in Company Bankruptcy Prediction

1st Xinyuan Sun
Questrom School of Business
Boston University
Boston, MA
xinyasun@bu.edu

2nd Shi Bo
Questrom School of Business
Boston University
Boston, MA
shibo@bu.edu

3rd Minheng Xiao
Questrom School of Business
Boston University
Boston, MA
minhengx@bu.edu

***Abstract*—Effective bankruptcy prediction is essential for money establishments to form applicable disposition decisions. In general, the input variables, financial ratios, and prediction techniques, such as machine learning, are the most significant factors poignant the prediction performance. whereas several connected works have projected novel prediction techniques, only a few have analyzed the discriminatory power of the features involving bankruptcy prediction. In this research report, we will utilize various machine learning methods to predict whether a company has bankrupted and compare them in many aspects.**

Contents

I. Introduction

Bankruptcy and business failures can have negative impacts each on the enterprises themselves and also the international economy. Business practitioners, investors, governments, and educational researchers have long studied ways in which to spot the potential risk of business failure so as to cut back the economic loss caused by bankruptcy. The main goal of this project is to use several basic machine learning techniques such as, Logistic Regression, Support Vector Machine(SVM), XGBoost etc, to predict whether corporations with several accounting indicators will bankrupt or not and compare the f1-scores, accuracy scores and AUC scores of different models. We analyze the situations where the model will perform better or worse. We also identify the reasons why some outlier companies are correctly classified and some are not.

II. Data

A. Data Acquisition and Prepossessing

The data is gathered from the Taiwan Economic Journal for the years 1999 to 2009 from Kaggle competition and is consist of ninety five features and one response variable: bankrupt or not. Features are essential accounting indicators, which includes ROA, P/E, ROA and so on. What we can image is that there are too many features which may cause potential problems, such as correlations between features, and over-fit, needed to be observed and tackled.

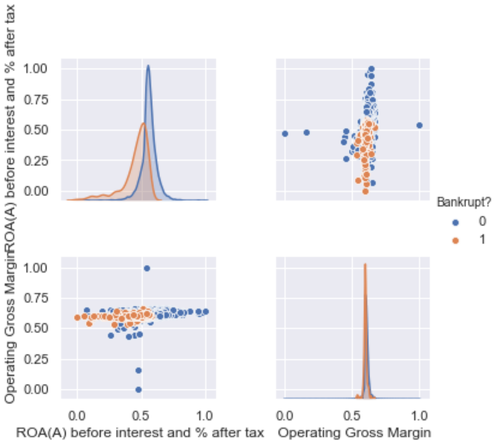

Fig. 1. Example of Imbalance Data

B. Potential Problems

The underlying issues may exist in the dataset including the following:

Identify applicable funding agency here. If none, delete this.

- Too many predictive variables
- Imbalanced classes
- Correlated variables
- Latent outliers
- Different feature magnitude

We will try to figure out them at the beginning and intermediate of our exploration, but first, we are supposed to detect if these problems truly embedded in the dataset, but let go deeply into the machine learning models first.

III. Machine Learning Methods Comparison and Exploration

We choose three basic machine learning classifiers including Logistic Regression, Support Vector Machine and XGBoost. The reason why we choose these three classifiers are that, since bankruptcy prediction is a bivariate classification problem, Logitsic Regression is suitable for this kind of problem and is easy to implement. Also, support vector machine classifier has variate kernels that can be used to solve different problems, and with several regularization parameters, it is more likely to fit better and perform better. Besides, as a resemble classifier, XGBoost can handle imbalanced problem more properly and is extremely different from methods such as Logistic Regression and Support Vector Machine we used.

There are several other reasons that we choose these three classifiers. In order to deliver those reasons, we summarize the situations that one model may perform better than others in the following table("*" means works better).

	SVM	LR	XGboost
High dimensional data	*		*
Not well separated data			*
Imbalanced data			*
Classification	*	*	*
Regularization	*	*	*
Much Computing power	*		*
Speed			*
Non-linearity data	*		*
Linearity data	*	*	*
Missing Values			*

A. Support Vector Machine

Support Vector Machine (SVM) is one of the most popular Machine Learning Classifiers. It falls under the category of Supervised learning algorithms and uses the concept of Margin to classify between classes.

In general the equation for a hyperplane has the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$. In two dimensions a hyperplane is a line and If β_0 = 0, the hyperplane goes through the origin, otherwise not.The vector β = (β1 ,β2 ,··· ,βp) is called the normal vector — it points in a direction orthogonal to the surface of the hyperplane.

Also, we can expand this to higher dimension with different kernel. Different kernel can deal with different structure of data (linear or non-linear). In our case, we used 4 kernels for the non-linear data.

1) SVM Results: We will represent the SVM results with different kernels in table 1 and the data has been standardized but not extract the principle components and remove outliers. The charts below are the out-of-sample result diagnosis which display low F1-scores for bankrupted companies. The model only predicted correctly six of the whole bankrupted companies even if the ROC curve looks well. However, since our classes are extremely imbalanced, ROC curve could not say loudly here.

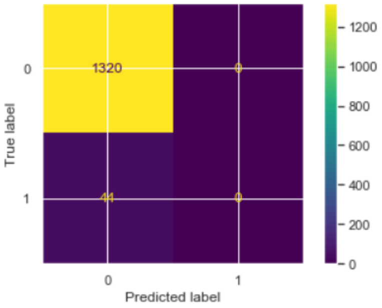

Fig. 2. SVM Out-of–Sample Confusion Matrix of RBF Kernel

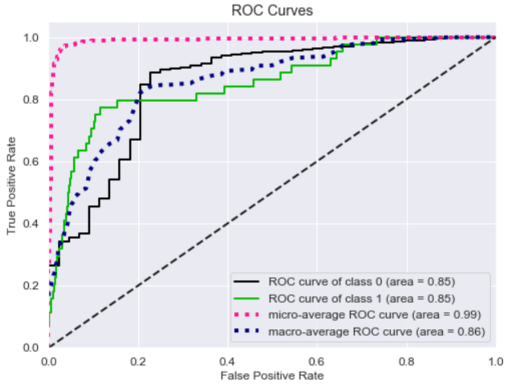

Fig. 3. SVM Out-of–Sample ROC curve of RBF Kernel

Table 1: Results of SVM

Kernel	F1-scores(Label 1)	AUC	Accuracy
RBF	0.000	0.854	0.967
Polynomial	0.128	0.857	0.968
Linear	0.241	0.897	0.967
Sigmoid	0.128	0.890	0.969

It is surprised that, empirically speaking, the RBF could work best in this case because the data may non-linear, but the best kernel here is "linear" which has 0.241 f1-scores for label 1. So, we will use linear kernel as the best kernel to do the following research. Also, we guess that SVM is superior to logistic regression since it has more regularization parameters and is more suitable for high dimensional data and more powerful computing ability.

B. Logistic Regression

Logistic regression is an appropriate regression analysis model to conduct when the dependent variable is dichotomous (binary). Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logistic regression uses the form:

$$p(X) = \Pr(Y = 1 \mid X)p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

No matter what values β_0, β_1 or X takes, p(X) belongs to (0,1).

Lightly speaking, logistics regression is a method of regression where we have a binary response variable; that is, a 1/0 response variable. More specifically, it predicts the probability that an observation is either a 1 or a 0. In this problem, bankrupt is treated as 1 and not bankrupt is treated as 0. We think logistic regression may fit this problem, but we didn't expect that it will have a good performance since out training data is imbalanced.

1) Logistic Regression Results: It also has different parameters that we could choose to improve our classifying results, for example, we could imply penalty to the model to preclude the model to be much over-fitting or we may determine disparate threshold to decide whether the company should be categorized as bankrupted or not, but usually we choose 0.5 as the threshold. The following table shows the performance of logistic regression:

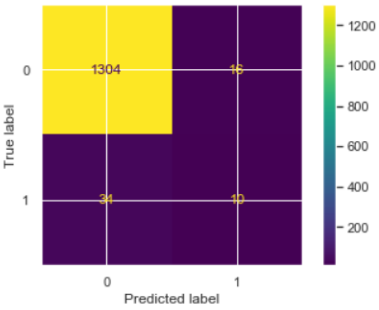

Fig. 4. Logit Out-of-Sample Confusion Matrix with L2-norm

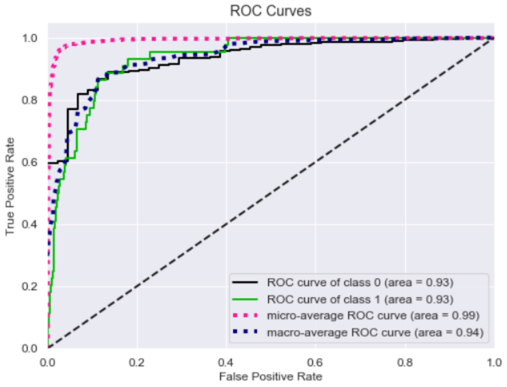

Fig. 5. Logistic Regression ROC curve

Table 2: Results of Logistic Regression

Panelty	F1-scores(Label 1)	AUC	Accuracy
L1-norm	0.320	0.912	0.962
L2-norm	0.286	0.934	0.963

In general, we might assume that logistic regression can be worse than SVM because it constructs linear boundaries and the main limitation of logistic regression is the assumption of linearity between the dependent variable and the independent variables. However, what is interesting here is that the L1-norm Logistic Regression actually performs better than SVM with the linear kernel. Hence, we should use the L1-norm logit to make the final comparison.

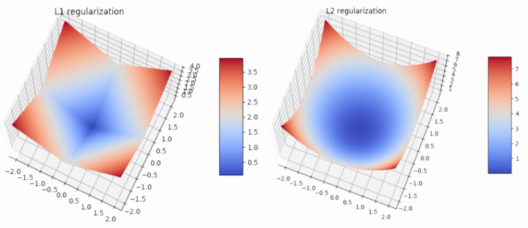

Fig. 6. L1 and L2 Penalty

We also explore why L1-norm logit performs better than L2-norm. L1 regularization adds an L1 penalty equal to the absolute value of the magnitude of coefficients. In other words, it limits the size of the coefficients. L1 can yield sparse models (i.e. models with few coefficients). Some coefficients can become zero and eliminated. L2 regularization adds an L2 penalty equal to the square of the magnitude of coefficients. L2 will not yield sparse models and all coefficients are shrunk by the same factor (none are eliminated). Therefore, by observing the properties between L1 and L2, we can conclude that The L1-norm prefers sparse coefficient vectors. This means the L1-norm performs feature selection and can delete all features where the coefficient is close to zero. A reduction of the dimensions is useful in most cases. It coincidentally tackled the main problem in dataset which is high dimensional. And this is the reason we conjecture that L1-norm Logistic Regression performs better than SVM with linear kernel.

C. XGBoost

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predicts the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

1) XGboost Results: We can simply conjecture that XGBoost may be the best model to utilize, since it has lots of advantages such as high flexibility, suitability for imbalanced data, and etc.

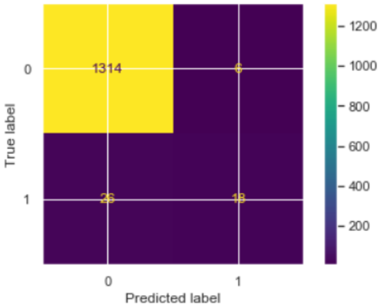

Fig. 7. XGboost Out-of-Sample Confusion Matrix with 1000 estimator

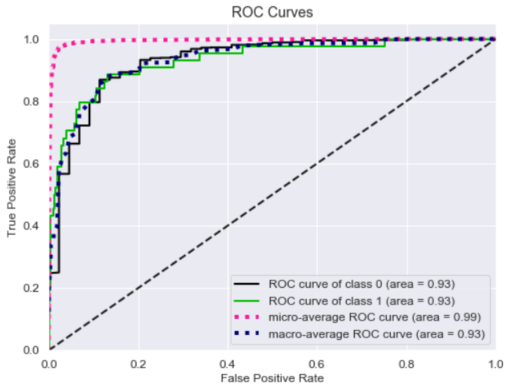

Fig. 8. XGboost ROC curve with 1000 estimator

Table 3: Results of XGBoost

$\mathrm{N}_Estimators$	F1-scores(Label 1)	AUC	Accuracy
10	0.290	0.886	0.967
100	0.393	0.938	0.972
1000	0.529	0.931	0.976
5000	0.406	0.927	0.972

We can see that when we choose number of boosting rounds(a.k.a. the maximum number of trees used) equals to 1000, it hits the best f1-score. The 5000 estimators does not function better than 1000, the reason probably is that too many estimators results in over-fitting.

To summarize this part, we make a comparison among these three classifiers.

Table 4: Comparison of three models

ML	F1-scores	AUC	Accuracy
SVM	0.290	0.886	0.967
Logistic Regression	0.393	0.938	0.972
XGboost	0.529	0.931	0.976

IV. SMOTE, Outliers Detection and PCA

In this final part, we will implement outliers detection technique to find out outliers in our dataset and improve our models' performance.

A. Imbalanced Learning

SMOTE is an oversampling technique that generates synthetic samples from the minority class. It is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the classifier.

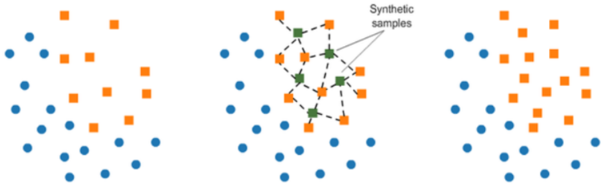

Fig. 9. SMOTE

B. Outliers Detection Method

The Local Outlier Factor is based on a concept of a local density, where locality is given by K-nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers.

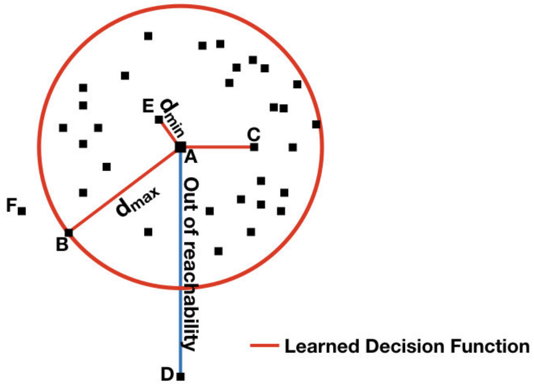

Fig. 10. Local Outlier Factor

The advantage of LOF is that a point will be considered as an outlier if it is at a small distance to the extremely dense cluster. The global approach may not consider that point as an outlier. But the LOF can effectively identify the local outliers.

On the other hand, the drawback of LOF is that since LOF is a ratio, it is tough to interpret. There is no specific threshold value above which a point is defined as an outlier. The identification of an outlier is dependent on the problem and the user.

C. Principle Component Analysis

PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data. The i-th principal component can be taken as a direction orthogonal to the first i-1 principal components that maximizes the variance of the projected data.

D. Results Comparsion

In this part, we combine all techniques we discussed above. Moreover, we tried to utilize and combine different methods and compare results with our best model and parameters. In the first circumstance, we implemented only PCA and SMOTE, the performance is shown in table 4.

Table 5: Results of PCA + SMOTE

ML	F1-scores	AUC	Accuracy
SVM	0.290	0.878	0.884
Logistic Regression	0.308	0.882	0.891
XGboost	0.353	0.903	0.951

Then we applied LOF technique to idenity outliers and then remove x percent of outliers in the train dataset. This percentage is calculated by the ranks of negative outlier factor.

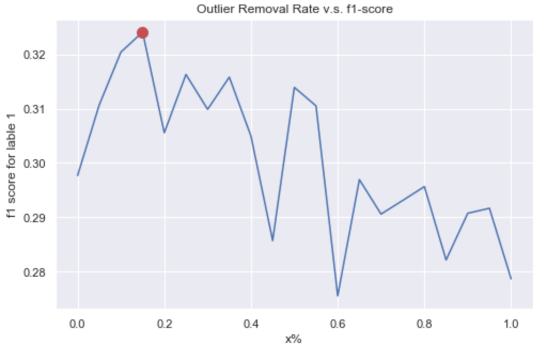

Fig. 11. x percent outliers removal for logistic regression

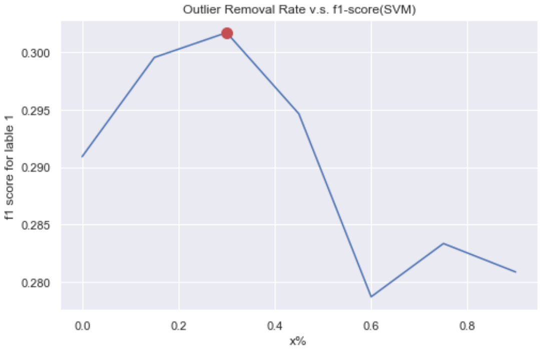

Fig. 12. x percent outliers removal for SVM

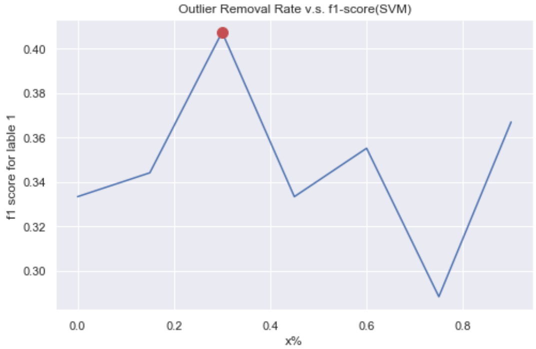

Fig. 13. x percent outliers removal for XGboost

We listed various percentiles of outliers we discarded for models, then we chose the best cutoff according to f1-score for label 1. As the figures above show that we can not just simply remove all the outliers in dataset since it will not provide us with the best performance.

After deciding the proper cutoffs, the performance of three classifiers is shown in Table 6.

Table 6: Results of PCA + SMOTE + Outlier Detection

ML	F1-scores	AUC	Accuracy
SVM	0.290/0.301	0.878/0.905	0.884/0.884
LR	0.308/0.326	0.882/0.908	0.891/0.893
XGboost	0.353/0.408	0.903/0.870	0.951/0.957

We can clearly observe that the performance improved a little bit higher than the original models with only PCA and SMOTE.

Finally, we try to extract the outlier companies and explore the reasons why they were misclassified. We select the best model which is XGBoost and use it to predict the outliers. We find that there are about 18 outliers which are misclassified. We divided all outliers into two categories and focused on the outliers in bankrupt category.

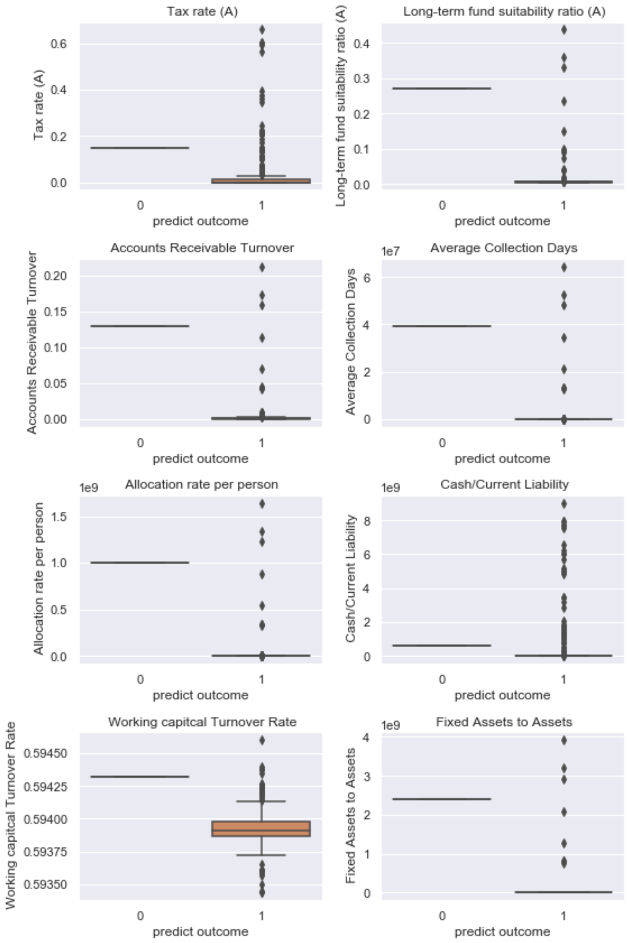

Fig. 14. Features Selected

We draw box plots for all features in order to identify the differences in the distribution of each feature of correctly predicted outliers and misclassified outliers. Based on our observations, we select eight features whose distributions are extremely different in correct and wrong prediction categories. Specifically, we can clearly see that misclassified outliers are basically lie in the area which drastically deviated from the correctly classified ones.

The features we think may have big influences are: Tax Rate , Long-term Fund Suitability Ratio ,Accounts Receivable Turnover, Average Collection Days, Allocation Rate Per Person, Cash/Current Liability, Working Capitcal Turnover Rate, Fixed Assets to Assets. We conclude these features are the reasons that the model misclassified these outliers companies.

V. Conclusion

In this paper, we introduce, compare models we use to predict bankruptcy for a first intuition. We also analyze the situations under which different machine learning models may perform better or worse. We also implement variate data preprocessing techiniques such as PCA and Standardization, outlier detection techinique such as LOF and imbalance learning method such as SMOTE. After applying these techniques, we find that the model performance gets better.We also find that the optimal cutoff for outliers is arount 15% to 20%. Finally, we extract the outlier companies and analyze main factors that resulting misclassification problems and we conclude that these factors include Tax Rate , Long-term Fund Suitability Ratio ,Accounts Receivable Turnover, Average Collection Days, Allocation Rate Per Person, Cash/Current Liability, Working Capitcal Turnover Rate, Fixed Assets to Asset.

References

Liang, D., Lu, C.-C., Tsai, C.-F., and Shih, G.-A. (2016) Financial Ratios and Corporate Governance Indicators in Bankruptcy Prediction: A Comprehensive Study. European Journal of Operational Research, vol. 252, no. 2, pp. 561-572. https://www.sciencedirect.com/science/article/pii/S0377221716000412

M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, ”Support vector machines,” in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

Peng Li, Siben Li, Tingting Bi and Yang Liu, ”Telecom customer churn prediction method based on cluster stratified sampling logistic regression,” International Conference on Software Intelligence Technologies and Applications International Conference on Frontiers of Internet of Things 2014, 2014, pp. 282-287, doi: 10.1049/cp.2014.1576.

J. Bao, ”Multi-features Based Arrhythmia Diagnosis Algorithm Using Xgboost,” 2020 International Conference on Computing and Data Science (CDS), 2020, pp. 454-457, doi: 10.1109/CDS49703.2020.00095.

Appendix

List of Figures

List of figures for ROC curve and confusion matrix of outliers removed results

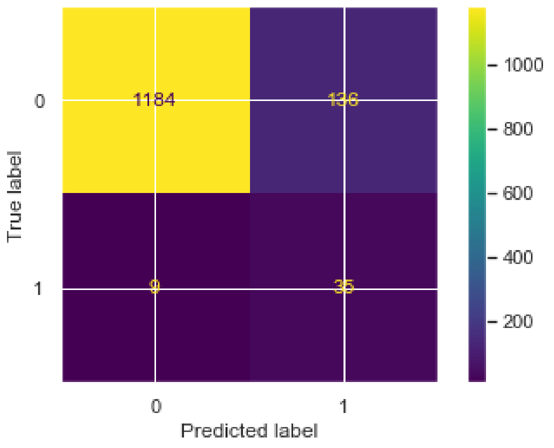

Fig. 15. Logistic Regression Confusion Matrix

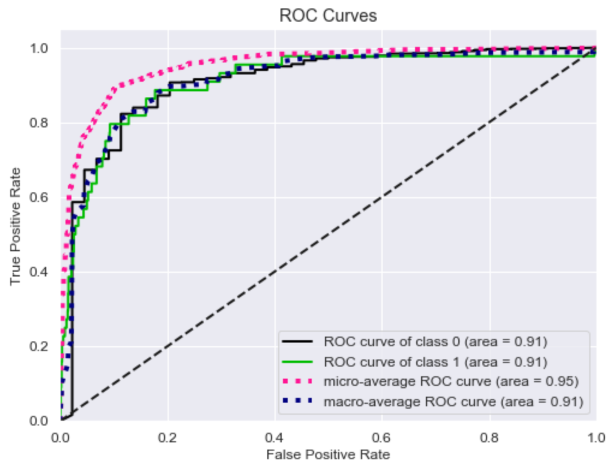

Fig. 16. Logistic Regression ROC curve

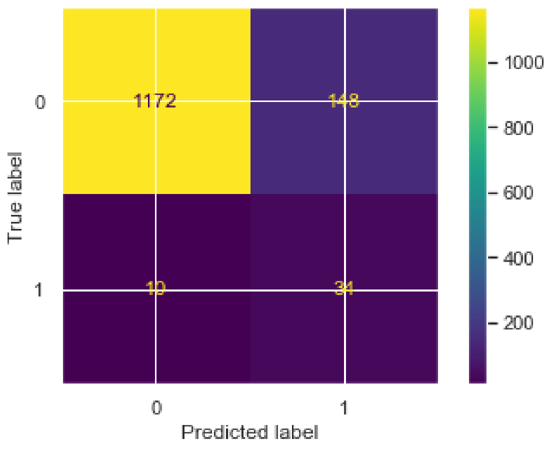

Fig. 17. SVM Confusion Matrix

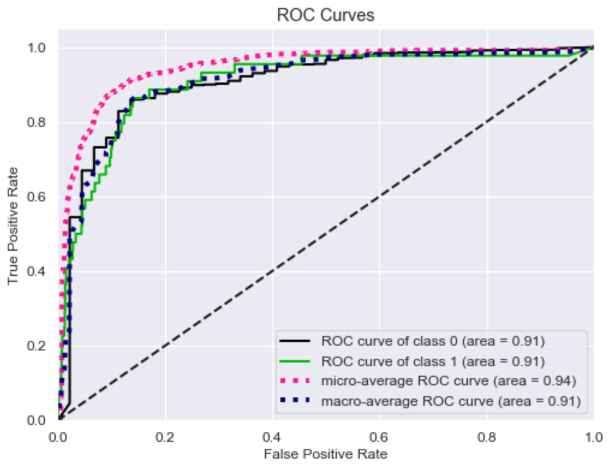

Fig. 18. SVM ROC curve

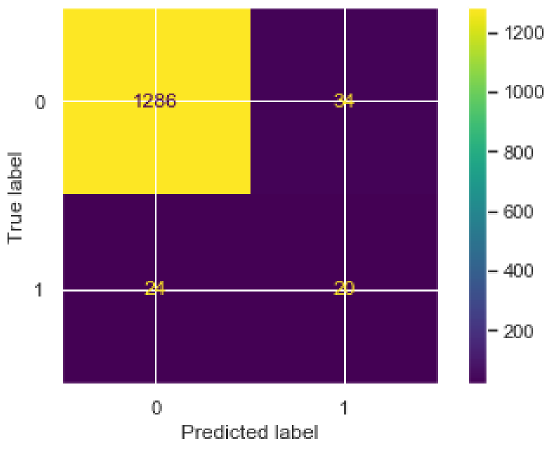

Fig. 19. XGBoost Confusion Matrix

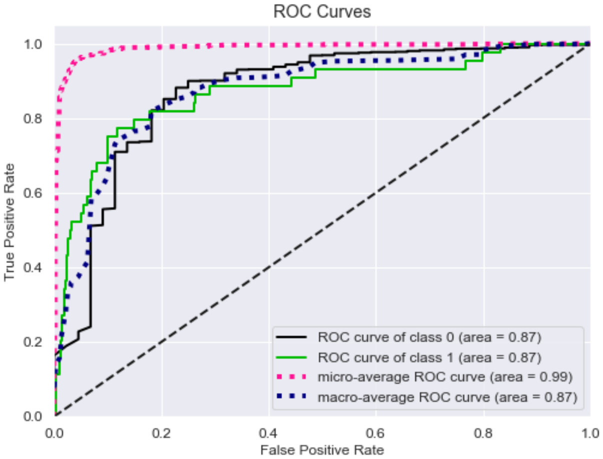

Fig. 20. XGBoost ROC curve