Machine Learning and Big Data

DATA 622

CUNY

School of Professional Studies

Professor Joseph Sabelja

Date: Nov 13, 2022

(HW3)

Exploratory Analysis and Essay

pprepared by

Anjal Hussan

Table of Contents

[Introduction 2](#_Toc119427865)

[Data 2](#_Toc119427866)

[Exploratory Data Analysis 2](#_Toc119427867)

[Correlations 8](#_Toc119427868)

[Split Data into Train and Test Data set 11](#_Toc119427869)

[Random Oversampling and Undersampling for Imbalanced Classification 12](#_Toc119427870)

[Decision Tree Classifier 13](#_Toc119427871)

[Random Forest 16](#_Toc119427872)

[KNN 17](#_Toc119427873)

[SVM 18](#_Toc119427874)

[Results and Discussion 20](#_Toc119427875)

[Conclusion 23](#_Toc119427876)

[Reference 24](#_Toc119427877)

# Introduction

Data Analysis is the statistics and probability to figure out trends in the data set. It helps in drilling down the information, to transform metrics, facts, and figures into initiatives for improvement. In this essay, we will explore the data using python and the supporting libraries.

# Data

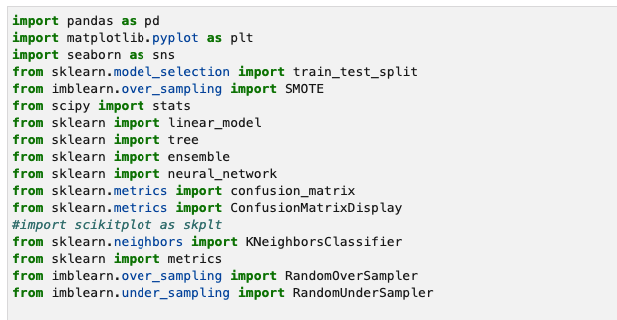
The data set we are going to use is the Taiwanese Bankruptcy Prediction Data Set publicly available in the UCI Machine earning Repository (<https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction>)

The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange

# Exploratory Data Analysis

We will explore the data sets and perform the exploratory data analysis in python.

First, we start by importing libraries in the Python notebook:



Once we import the libraries and load the csv files, we explore the shape of the data frame:

Graphical user interface, application

Description automatically generated

Let’s take a look to the head of the dataset. For our screenshot, we are going to display first few columns of the data set.

Table

Description automatically generated

Is there any null value in the data frame?

Graphical user interface, text, application, chat or text message

Description automatically generated

Let’s find out the name of the columns in this data frame:

Graphical user interface, text, application

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Bankrupt? | Operating Gross Margin | After-tax net Interest Rate |
| ROA(C) before interest and depreciation before interest | Realized Sales Gross Margin | Non-industry income and expenditure/revenue |
| ROA(A) before interest and % after tax | Operating Profit Rate | Continuous interest rate (after tax) |
| ROA(B) before interest and depreciation after tax | Pre-tax net Interest Rate | Operating Expense Rate |
| Research and development expense rate | Net Value Per Share (A) | Operating Profit Per Share (Yuan ¥) |
| Cash flow rate | Net Value Per Share (C) | Per Share Net profit before tax (Yuan ¥) |
| Interest-bearing debt interest rate | Persistent EPS in the Last Four Seasons | Realized Sales Gross Profit Growth Rate |
| Tax rate (A) | Cash Flow Per Share | Operating Profit Growth Rate |
| Net Value Per Share (B) | Revenue Per Share (Yuan ¥) | After-tax Net Profit Growth Rate |
| Regular Net Profit Growth Rate | Cash Reinvestment % | Debt ratio % |
| Continuous Net Profit Growth Rate | Current Ratio | Net worth/Assets |
| Total Asset Growth Rate | Quick Ratio | Long-term fund suitability ratio (A) |
| Net Value Growth Rate | Interest Expense Ratio | Borrowing dependency |
| Total Asset Return Growth Rate Ratio | Total debt/Total net worth | Contingent liabilities/Net worth |
| Operating profit/Paid-in capital | Average Collection Days | Operating profit per person |
| Net profit before tax/Paid-in capital | Inventory Turnover Rate (times) | Allocation rate per person |
| Inventory and accounts receivable/Net value | Fixed Assets Turnover Frequency | Working Capital to Total Assets |
| Total Asset Turnover | Net Worth Turnover Rate (times) | Quick Assets/Total Assets |
| Cash/Total Assets | Revenue per person | Current Assets/Total Assets |

For the sake of simplicity, some of the variable names have been dropped from the above table.

Let’s look at the column Bankrupt? Because this column is one of the columns that we are going predict based on the other variables.

Chart

Description automatically generated

Since the dataset contains 96 variables and a lot of the columns we are not going to user in our training, let’s make a new data frame using the variables that will be in our interest to predict the target variable.

Table

Description automatically generated

Now let’s look at the distribution of the data from our data frame

Text

Description automatically generated

Chart

Description automatically generated

Chart, histogram

Description automatically generated

Chart

Description automatically generated

Chart, line chart

Description automatically generated

# Correlations

The correlations between variable are very important to draw any conclusion regarding the relationship between data points. Looking at the row data, it’s hard to find any relationship however, looking at the visual, it’s obvious to establish a correlation. Let’s construct correlation matrix to observe the strength of relationships of each variable with bankruptcy.

Icon

Description automatically generated with medium confidence

From the distribution and the heatmap, there are a lot of multicollinearity issues, skewed features and the data is imbalanced.

Let’s look at the relationship between some variables:

Does Net Income to Total Assets has any relationship with being bankrupt?

Chart

Description automatically generated

Does Retained Earnings to Total Assets has any relationship with being bankrupt?

Chart, histogram

Description automatically generated

Does Total Asset Growth Rate have any relationship with being bankrupt?

Chart, histogram

Description automatically generated

# Split Data into Train and Test Data set

Now that we have explored the data, we want to come up with a model which should predict the target variable based on the data provided. in our case the target variable is “Bankrupt?”

Using the sklearn library, we now divide the data set into training and test data set

Graphical user interface, text, application

Description automatically generated

# Random Oversampling and Undersampling for Imbalanced Classification

As we already seen that the data is imbalanced and skewness exists in the dataset, we need to address this issue. The skewness in training dataset can influence the modes leading some to ignore the minority class entirely. To address this issue, we need to randomly resample the training dataset. The two approaches are as follows:

* Undersampling
* Oversampling

We will take advantage of the library imbalanced-learn available.

Graphical user interface, text, application, email

Description automatically generated

Now that we have two additional training set, let’s draw a baseline accuracy which will help us with our model evaluation

Graphical user interface, text, application

Description automatically generated

# Decision Tree Classifier

Using our training datasets, let’s make Decision tree classifier:

Graphical user interface, text, application

Description automatically generated

In the above code, we essentially created three models. First one is the regular Decision tree classifier model which took our original training datasets. The second and third took the undersampling and oversampling training dataset respectively.

Now that we have the models, let’s evaluate with our test dataset:

Graphical user interface, text, application

Description automatically generated

Looks like the undersampling model performance is not near the baseline. Oversampling model is close to the baseline buy none of the models was able to beat the base line in terms of their accuracy.

Let’s take a look which variable made most impact in terms of its importance.

Chart, histogram

Description automatically generated

Confusion matrix of the oversampling decision tree model:

Chart

Description automatically generated

In the confusion matrix, the count of wrong label is 71.

Confusion matrix of the undersampling decision tree model:

Chart, treemap chart

Description automatically generated

In the confusion matrix, the count of wrong label is 275.

Confusion matrix of the regular decision tree model:

Chart, treemap chart

Description automatically generated

In the confusion matrix, the count of wrong label is 125.

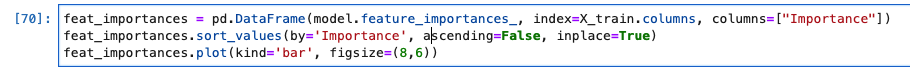
# Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Let’s apply the Random Forest classifier to predict our target variable.

Graphical user interface, text, application

Description automatically generated

Let’s take a look which variable made most impact in terms of its importance in the Random Forest model.



Chart, bar chart

Description automatically generated

Confusion Matrix for the Random Forest Model:

Chart, treemap chart

Description automatically generated

In the confusion matrix from Random Forest Model, the count of wrong label is 61.

# KNN

KNN is one of the simplest forms of machine learning algorithms mostly used for classification. It classifies the data point on how its neighbor is classified. KNN classifies the new data points based on the similarity measure of the earlier stored data points. Let’s apply the KNN algorithms to predict our target variable.

Graphical user interface, text, application, email

Description automatically generated

Confusion Matrix for the KNN:

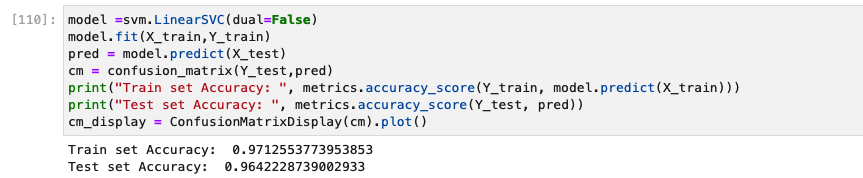
Chart

Description automatically generated

In the confusion matrix from KNN Model, the count of wrong label is 62.

# SVM

A support Vector machine is a supervised machine learning model that is used in both classification and regression problems. But SVM mostly gets used in classification problems. The SVM algorithm takes each data point and put it in an n-dimensional space. n is the number of features available in the data set. The value of each feature is the coordinate of the data point. Using this algorithm SVM solves the classification problem by finding the hyper-plane that differentiates the classes. Let’s apply the SVM algorithms to predict our target variable.



Confusion Matrix for the SVM:

Chart, treemap chart

Description automatically generated

In the confusion matrix from svm.LinerSVC Model, the count of wrong label is 61. The accuracy is 96.42% also close to the baseline accuracy.

We also created a SVM model with Linear kernel and code looks like this:

Graphical user interface, text, application

Description automatically generated

The confusion matrix for the svm.svc is below:

Chart, treemap chart

Description automatically generated

In the confusion matrix from svm.svc Model, the count of wrong label is 61. The accuracy is 96.42% also close to the baseline accuracy.

# Results and Discussion

In this study, seven alterative models were build using the relevant training data set. To evaluate these models, we used the remaining data set in the test data set for each model. the accuracy rates for each model are given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Classifier Model** | **Description** | **Accuracy Rate** | **Wrong Label** |
| Decision Tree Classifier | Regular Decision Tree Model | 0.9267 | 125 |
| Undersampler | 0.8387 | 275 |
| Oversampler | 0.9584 | 71 |
| Random Forest | RandomForestClassifier, random\_state=1 | 0.9642228739002933 | 61 |
| KNN | KNeighborsClassifier, n\_neighbors=2 | 0.9636363636363636 | 62 |
| Support Vector Machine | LinearSVC | 0.9642228739002933 | 61 |
| SVC | 0.9642228739002933 | 61 |

Accuracy rates were used to describe the usefulness of the models. Accuracy is probably the most used metric to measure the performance of targeting models in classification applications.

The confusion Matrix for decision tree model doesn’t look very good because of the higher count of wrong labels. Rest of the models produces similar kind of confusion matrix with the count of wrong label is around 61.

**Performance Evaluation Metrics:**

Let’s look to the ROC curve (AUC) of the models. Since the Decision Tree models are poor models, we are not going to consider the performance of the Decision Tree models. To evaluate the performance metrics, I used the area under the ROC Curve (AUC) which was computed using continuous outputs of the classifiers (distances from separating hyperplane for SVMs and outcome probabilities for RFs).

Let’s look at the AUC for each of the models to evaluate their performances:

|  |  |  |
| --- | --- | --- |
| **Classifier Model** | **Description** | **AUC** |
| Random Forest | RandomForestClassifier, random\_state=1 | 0.5157 |
| KNN | KNeighborsClassifier, n\_neighbors=2 | 0.5233 |
| Support Vector Machine | LinearSVC | 0.5236 |
| SVC | 0.5 |

Visualize the AUCs

|  |  |
| --- | --- |
| KNN | Random Forest  Chart, line chart  Description automatically generated |
| LinearSVC  Chart, line chart  Description automatically generated | SVC  Chart, line chart  Description automatically generated |

The AUC of all the models is around 0.5. The value for AUC ranges from 0 to 1. A model that has an AUC of 1 can perfectly classify observations into classes. The higher the AUC score, the better the model can classify the observations into classes. However, there is no magic number that determines the good or bad AUC score.

Hosmer and Lemeshow described the rule of thumb in Applied Logistic Regression book and created a chart to understand AUC Score. The chart is as follows:

* 0.5 = No discrimination
* 0.5-0.7 = Poor discrimination
* 0.7-0.8 = Acceptable discrimination
* 0.8-0.9= Excellent discrimination
* >0.9 = Outstanding discrimination

The chart indicates that the performance of our models is not excellent but it’s not poor either.

In general, both Random Forest and SVM works well for binary classification. In the classification problem, Random Forest gives us the probability of belonging to class, whereas SVM gives us the distance to the boundary. Where SVM applies, it generally performs better than Random Forest but, in our case, we don’t see any significant difference between SVM and RF models. The choice of algorithm depends upon the desired outcome. Although both models are good at their place, but it very much depends upon the quality of data when it comes to algorithm’s performance.

# Conclusion

Among the models, it’s seeming to me that Random Forest model was closest to the baseline in terms of the test data set accuracy which is 0.9642. the count of incorrect predicted label is also lowest. Hence, we can conclude that our best model is the Random Forest Model to predict the target variable for the Taiwanese Bankruptcy Prediction Data Set.

# Reference

Applied Logistic Regression, 3rd Edition (3rd ed.). (2013). Wiley.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. Journal of Thoracic Oncology, 5(9), 1315–1316. https://doi.org/10.1097/jto.0b013e3181ec173d

Heumann, B. W. (2011). An Object-Based Classification of Mangroves Using a Hybrid Decision Tree—Support Vector Machine Approach. Remote Sensing, 3(11), 2440–2460. <https://doi.org/10.3390/rs3112440>

Ahmad, A., Safi, O., Malebary, S., Alesawi, S., & Alkayal, E. (2021). Decision Tree Ensembles to Predict Coronavirus Disease 2019 Infection: A Comparative Study. *Complexity*, *2021*, 1–8. https://doi.org/10.1155/2021/5550344

Guhathakurata, S., Kundu, S., Chakraborty, A., & Banerjee, J. S. (2021). A novel approach to predict COVID-19 using support vector machine. *Data Science for COVID-19*, 351–364. https://doi.org/10.1016/b978-0-12-824536-1.00014-9

Alexander Statnikov, & Constantin F. Aliferis. (2007). Are random forests better than support vector machines for microarray-based cancer classification? *American Medical Informatics Association Annual Symposium*. https://pubmed.ncbi.nlm.nih.gov/18693924/