**Course Project of Introduction to Machine Learning**

Instructor: Dr. Louis Plebani

Ran Ran [rar418@lehigh.edu](mailto:rar418@lehigh.edu)

Yuan Yao [yuy418@lehigh.edu](mailto:yuy418@lehigh.edu)

2018.12.15

**Introduction**

Machine learning is the study of algorithms and statistical models that computer systems use to progressively improve their performance on a specific task. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. It is closely related to computational statistics, which focuses on making predictions using computers.

**Abstract**

In the project, knowing the features of the bank clients such as the age, job, marital, we are going to build a classification model to predict whether a client will subscribe or not. In the given data, we already know the final decision of these clients and we trained our model by using these data.

Firstly, since the data is imbalanced, we should resample the minority data to make the ratio to be near 1:1, then we transform the categorical data to numerical data, since model should use numerical data. After these 2 steps, we get a resampled numerical data set. On the top of that, we use resampled numerical data to fit the following models: Logistic Regression, K Nearest Neighbors, Random Forest, Support Vector Machine and Neural Network.

Finally, we use the model which performs best to do the prediction for future data.

**Experimental details**

First of all, we import the data.csv and future.csv, then preparing the data we need to process.

**1 Transform the data type:**

We found that the not all the data is numerical, some of the variable such as job, has a feature like ’housemaid’ or ‘services’. We have to turn it into numerical data. So we Used the label encoder function and successfully transform the categorical data to numerical data.

**2 Check the y distribution:**

After drawing a count plot of the data, we found that the data is imbalanced, the ratio of client who do not subscribe to who subscribe is about 9:1. In this case we need handle the imbalanced data first.

**3 Resampling the data**

We import resample from sklearn.utils. By using the resample function, we randomly duplicate ‘yes’ data from existing data so that the ratio of ‘no’ to ‘yes’ will be nearly 1:1.

**4 Training better classification model**

In the fitting part of our project, we split the data and standardized data into training data and test data. Test size is 0.20, and random state is 101. Then we train the logistic Regression, K Nearest Neighbors, Decision Tree and Random Forest, Support Vector Machine (Grid Search) and Neural Network respectively. Each experiment we get a prediction and a classification report.

**Content**

**0 Handle the imbalanced data**

If we using the imbalanced data to training the classification model, just using logistic regression as an instance:

precision recall f1-score support

0 0.91 0.99 0.95 7211

1 0.63 0.19 0.29 826

avg / total 0.88 0.91 0.88 8037

From the above report, we found the precision looks OK, but there are huge difference between 0 and 1 in their own precision, recall and f1-score. This shows that this model could only predict y=0 accurately, but not for 1 also.

As a result, we could not use imbalanced data to training model.

**1 Logistic Regression**

It is a simple yet powerful algorithm for binary classification, in this model, we import logisticRegression from the sklean.linear\_model and train the model. The following results are the accuracy of the prediction.

precision recall f1-score support

0 0.72 0.75 0.73 7245

1 0.74 0.70 0.72 7181

avg / total 0.73 0.73 0.72 14426

**2 K Nearest Neighbors**

It is a simple but useful algorithm. Its training algorithm is storing all the data, and its prediction algorithm are:

* Calculate the distance from x to all points in our data
* Sort the points in data by increasing distance from x
* Predict the majority label of the “K” closest points

In the experiment we get best result as setting n\_neighbors=K =1, The result under KNN is shown blow:

precision recall f1-score support

0 0.99 0.91 0.94 7245

1 0.91 0.99 0.95 7181

avg / total 0.95 0.95 0.95 14426

**3 Decision Tree and Random Forest**

This algorithm firstly draws a random (bootstrap) sample of size n (randomly choose n samples from the training set with replacement).

Then it Grow a decision tree from the bootstrap sample. At each node:  
1. Randomly select d features without replacement.  
2. Split the node using the feature that provides the best split according to the objective function, e.g., maximizing the information gain.

Then repeat the steps 1 to 2, …, k times.

After that it aggregate the prediction by each tree to assign the class label by majority vote.

In the experiment we set the K as 100 and get the result blow:

precision recall f1-score support

0 1.00 0.89 0.94 7245

1 0.90 1.00 0.95 7181

avg / total 0.95 0.94 0.94 14426

**4 Support Vector machine (Grid Search)**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), support vector machines (SVMs, also support vector networks) are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyze data used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-[probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier) [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier) (although methods such as [Platt scaling](https://en.wikipedia.org/wiki/Platt_scaling) exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

After doing Grid Search, we found better parameters for our SVM model. Here is the classification report for this model:

precision recall f1-score support:

0 1.00 0.99 0.99 7245

1 0.99 1.00 0.99 7181

avg / total 0.99 0.99 0.99 14426

**5 Neural Network**

Artificial neural networks (ANN) or [connectionist](https://en.wikipedia.org/wiki/Connectionism) systems are computing systems vaguely inspired by the [biological neural networks](https://en.wikipedia.org/wiki/Biological_neural_network) that constitute animal [brains](https://en.wikipedia.org/wiki/Brain). The neural network itself is not an algorithm, but rather a framework for many different [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms to work together and process complex data inputs.

In our project we have 15 dims data for input, using loss function= binary\_crossentropy, doing 500 epochs fitting. We get a model which classification report as below.

precision recall f1-score support

0 0.83 0.72 0.78 8340

1 0.68 0.80 0.74 6086

avg / total 0.77 0.76 0.76 14426

**Results discussions and Conclusions**

Actually, we got confusion matrix for these models’ result. For this Classification report:

Precision=TP/(TP+FP);

Recall=TP/(TP+FN);

F1-score=2\*TP/(2\*TP+FP+FN);

In this case, we should first choose the model which got both higher precision for 0 and 1, so we could not use the imbalanced data to train models. And from the results of all the models, we found SVM (Grid Search) get the best score for the classification. This model gets 0.99 precision for the classification problem.

We finally decide to use the SVM (Grid Search) model to do the predication for the future data.

**Reference**

How to write a technical report:

<https://www.aresearchguide.com/writing-a-technical-report.html>

How to handle imbalanced data before doing classification:

<https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/>

An introduction to Statistical learning:

<https://www-bcf.usc.edu/~gareth/ISL/>

**Appendix**

Projectfinal.py