Group 1 Project Description

Team Member:

Xiulin Hou, Jialei Huang, Ling Zhang

We will use Bank Market Data Set ([http://archive.ics.uci.edu/ml/datasets/Bank+ Marketing](http://archive.ics.uci.edu/ml/datasets/Bank+Marketing)) from UCI Machine Learning Repository. The data is related to market campaign, a way to promote a product through different media, including television, radio, print and online platforms. Businesses operating in high competitive markets may initiate frequent marketing campaigns and devote significant resources to generating brand awareness and sales.

The dataset includes features of potential clients and results of telemarketing calls for selling bank long-term deposits. Our classification goal is to predict whether a client will subscribe a term deposit. In this project, we will use software R, Python and potentially Data Robot. Python package *scikit-learn* can be used to to do preprocessing, exploratory data analysis, and modelling.

As this is a supervised learning, we plan to devolve 80% of data as training set and 20% for testing. We see over 75% accuracy as success.

However, this dataset has two main limitations: Firstly, it comes from a Portuguese retail bank, which cannot represent clients in other nations or from other banks. For this limitation, unfortunately, we haven’t found any other data for supplement. Secondly, the data is collected during financial crisis - between 2008 and 2013. This may affect result in a more negative tendency for client to accept the term deposit solicitation than under normal economic condition. For this limitation, we plan to adjust the depression with CPI, CCI, and unemployment rate data.

**Dataset Description:**

1. 16 input variables, both numeric and categorical, including age, job, marital status, education level, credit, contact communication type, house loan, personal loan, etc.
2. For categorical variables, we can use OneHotEncoder to preprocess.
3. 1 output variable: whether the client subscribed a term deposit (binary: ‘yes’, ‘no’), which should be translated to binary indicator with only value 0 and 1.
4. 45211 observations without any null value

There are 39922 response with answer “no” and 5289 response with answer “yes”. In order to deal with an imbalance dataset, we can add more samples or remove some samples by applying package *imbalanced-learn* after we split training and testing data.

**Data Preprocessing:**

1. Standardization: StandardScaler, MinMaxScaler
2. Imbalanced Data:
   1. Resampling:
      1. RandomOversampler: repeat samples from the minority class randomly
      2. RandomUnderSampler: drop data from the majority class randomly
   2. Class-weights:
      1. Similar for linear and non-linear SVM
      2. Can be used in linear and tree models
   3. Smart resampling:
      1. Edited nearest neighbors, condensed nearest neighbors, SMOTE (synthetic minority oversampling technique)

**Variable Selection:**

1. *SelectKBest*: select features according to the k highest scores (due to function *f\_classif, mutual\_info\_classif, SelectPercentile*)
2. PCA: use linear decomposition reduction to a lower dimensional space and use *n\_components* to tune parameter

**Model Selection:**

1. Cross Validation: *GridSearchCV* exhaustive search over specified parameter values for an estimator
2. Logistic Regression: to classify the result using sigmoid function
3. Naive Bayes: classify results by applying Bayes' theorem with strong (naive) independence assumptions between the features
4. SVM: a model that maps points into space, which is divided by a calculated clear gap that is as wide as possible
5. AdaBoost (AdaBoostClassifier): we can use this to improve other simple models. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier.
6. Random Forest (*RandomForestClassifier*): an ensemble method that constructs a multitude of decision trees at training time and outputting the class that is the mode of the classes
7. Gradient Boosting (GradientBoostingClassifier): produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees