

Drug Activity Prediction Using Adaboost

Abhishek Jallawaram, Ragnarok7417, 0.70,100

ajallawa@gmu.edu, G01373042

Abstract

This document provides process of handling imbalanced datasets using feature selection, dimensionality reduction and selecting the best classification model.

1. Introduction

This document covers implementation of Principal Component Analysis(PCA), Singular Value Decomposition(SVD), Chi Square, SelectKbest, Adaboost algorithm on an imbalanced dataset.

1.1. Objectives

Listed below are the objectives achieved in this assignment:

- Dimensionality Reduction techniques
- Handle imbalanced data (Drug Activity Prediction)
- Cross Validation
- Choose the best model i.e., Adaboost Algorithm with best metrics

2. Problem Scenario

A molecule can be represented by several thousands of binary features which represent their topological shapes and other characteristics important for binding. The dataset has an imbalanced distribution i.e., within the training set there are only 78 actives (+1) and 722 inactives (0). Determine whether a given particular compound is active (1) or inactive (0).

3. Data Engineering

The given data is converted into a binary sparse matrix and below methods are deployed.

- Chi Square with SelectKbest
- Principal Component Analysis(PCA)
- Singular Value Decomposition(SVD)
- Synthetic Minority Oversampling Technique (SMOTE)

3.1. Chi Square

Chi square is used to determine whether the data is independent or related to each other. Chi-square test measures dependence between stochastic variables which enables us to weed out features that are independent and irrelevant for classification.

3.2. SelectKbest

This technique is used with Chi square to select features based on 'k' highest scores.

3.3. Principal Component Analysis

Principal Component Analysis(PCA) is used to transform the data into a lower dimensional space, by constructing dimensions that are linear combinations of the input dimensions/features. Below steps are performed to reduce the dimensions using PCA.

- Standardization of each column in the row.
- Co-Variance Matrix Computation
- Compute Eigen values and Eigen vectors of co-variance matrix to identify the principal components
- Transformation of data based on the principal component axes.

In our scenario, we have used 0.95% as the metric for dimensionality reduction for PCA.

3.4. Singular Value Decomposition

Truncated SVD produces a matrix factorization where the number of columns can be specified for a number of truncation.Truncated SVD can deal with sparse matrix to generate features' matrices, whereas PCA would operate on the entire matrix for the output of the covariance matrix.

In our scenario we have used the n_components as 350 based on the variance value 55% with training data which has produced the best results in cross validation. The below values were used to test the variance validation. {100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800}

3.5. Synthetic Minority Oversampling Technique

SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random.We perform over-sampling using SMOTE to adjust the imbalanced data.

4. Model Selection

The data is loaded and converted into a sparse matrix. The train data is split into train and validation set. Data engineering steps are performed to the dataset and cross validation is performed to compare the metrics. AdaBoost stands for adaptive boosting which refers to combining multiple weak classifiers to a single strong classifier.

In our scenario we have used Decision Stump (Decision tree with single split), Decision Tree with max depth as 2, Naive Bayes and Logistic Regression as the base classifiers to build a single strong classifier.

The rows are added samples weights and they are selected on a probability of their weights. A classifier is built on the new sampled dataset and the weights are adjusted based on their accuracy. This process is repeated till the error becomes zero or the iterations are completed.

5. Obervation

Adaboost is performed on the below scenarios : SMOTE + Truncated SVD, SMOTE + PCA, SMOTE + Chi Square + SelectKbest, Truncated SVD, PCA, Chi Square + SelectKbest. The best metrics were provided when we have used the adaboost algorithm with decision stumps, chi square and selectKbest.

Classifier	Accuracy	Metrics
	Average	F1_MacroAvg
Adaboost (Decision Stump) + SMOTE + SVD	0.9000	0.7400
Adaboost (Decision Stump) + SMOTE + PCA	0.8400	0.6900
Adaboost (Decision Stump) + SMOTE + Chi Square	0.9400	0.8000
Adaboost (Decision Stump) + SVD	0.6800	0.5500
Adaboost (Decision Stump) + PCA	0.7600	0.5900
Adaboost (Decision Stump) + Chi Square	0.9800	0.9400
Adaboost (Logistic Regression) + SMOTE + SVD	0.8900	0.7600
Adaboost (Logistic Regression) + SMOTE + PCA	0.8400	0.6900
Adaboost (Logistic Regression) + SMOTE + Chi Square	0.9400	0.8200
Adaboost (Logistic Regression) + SVD	0.4200	0.3800
Adaboost (Logistic Regression) + PCA	0.2900	0.2800
Adaboost (Logistic Regression) + Chi Square	0.8900	0.7800

Fig 1 – Model Metrics

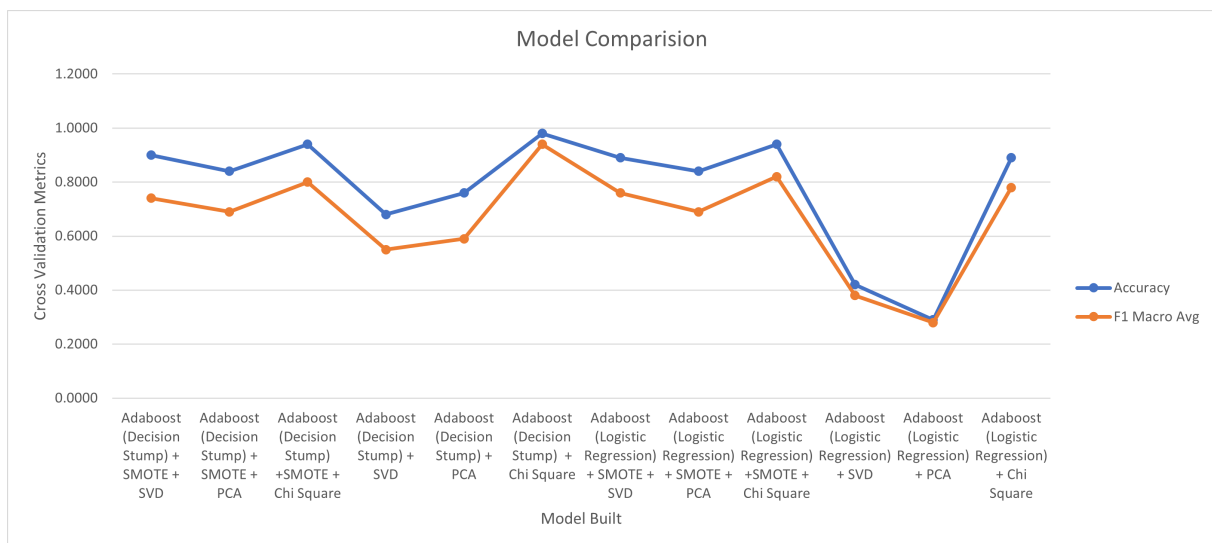


Fig 2 – Model Comparison

6. References

- TruncatedSVD=<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>
- PCA=<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- SMOTE=https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html