George Mason University
Department of Computer Science 584-006
Virginia,USA

Drug Activity Prediction

Abhishek Jallawaram, Ragnarok7417, 0.73,132
ajallawa@gmu.edu, G01373042

Abstract

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

1. Introduction

This document is covers implementation of Principal Component Analysis(PCA), Singular Value Decomposition(SVD), Chi Square, SelectKbest, Naive Bayes classifier and Decision tree classifier on 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., Naive Bayes/Decision Tree, F1 Score, Precision, Recall and Accuracy

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 in to 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. Naive Bayes classifier and Decision Tree classifier are used to determine the best model and prediction is performed on the test data.

5. Obervation

Naive Bayes classifier (GaussianNB,BernoulliNB) and Decision Tree classifier has been performed with below data engineering.

- SMOTE + Truncated SVD
- SMOTE + PCA
- Chi Square + SelectKbest
- Truncated SVD
- PCA

Classifier	Cross Validation - Accuracy						Metrics
	CV1	CV2	CV3	CV4	CV5	Average	F1_MacroAvg
Decision Tree + SMOTE + SVD	0.8906	0.8984	0.8672	0.8906	0.9375	0.8969	0.66
Naive Bayes : GaussianNB + SMOTE + SVD	0.7813	0.8516	0.8203	0.8906	0.8984	0.8484	0.55
Naive Bayes : BernoulliNB + SMOTE + SVD	0.9297	0.9453	0.8672	0.9063	0.9219	0.9141	0.82
Decision Tree + SMOTE + PCA	0.8984	0.9688	0.8750	0.8750	0.8828	0.9000	0.86
Naive Bayes : GaussianNB + SMOTE + PCA	0.7109	0.7734	0.7344	0.7891	0.8125	0.7641	0.09
Naive Bayes : BernoulliNB + SMOTE + PCA	0.9531	0.9453	0.8906	0.9141	0.9219	0.9250	0.86
Decision Tree + Chi Square	0.7785	0.7128	0.7405	0.7439	0.7682	0.7488	0.84
Naive Bayes : GaussianNB + Chi Square	0.7232	0.6055	0.6367	0.6367	0.6367	0.6478	0.73
Naive Bayes : BernoulliNB + Chi Square	0.7439	0.6817	0.7059	0.7301	0.7370	0.7197	0.85
Decision Tree + SVD	0.8828	0.9063	0.8516	0.8906	0.8906	0.8844	0.79
Naive Bayes : GaussianNB + SVD	0.8359	0.9063	0.8281	0.8828	0.8906	0.8688	0.59
Naive Bayes : BernoulliNB + SVD	0.9297	0.9609	0.8906	0.9531	0.9375	0.9344	0.84
Decision Tree + PCA	0.8828	0.8984	0.8828	0.8672	0.8984	0.8859	0.75
Naive Bayes : GaussianNB + PCA	0.6875	0.8203	0.7813	0.8672	0.8672	0.8047	0.5
Naive Bayes : BernoulliNB + PCA	0.9375	0.9609	0.9141	0.9453	0.9297	0.9375	0.8

Fig 1 – Model Metrics

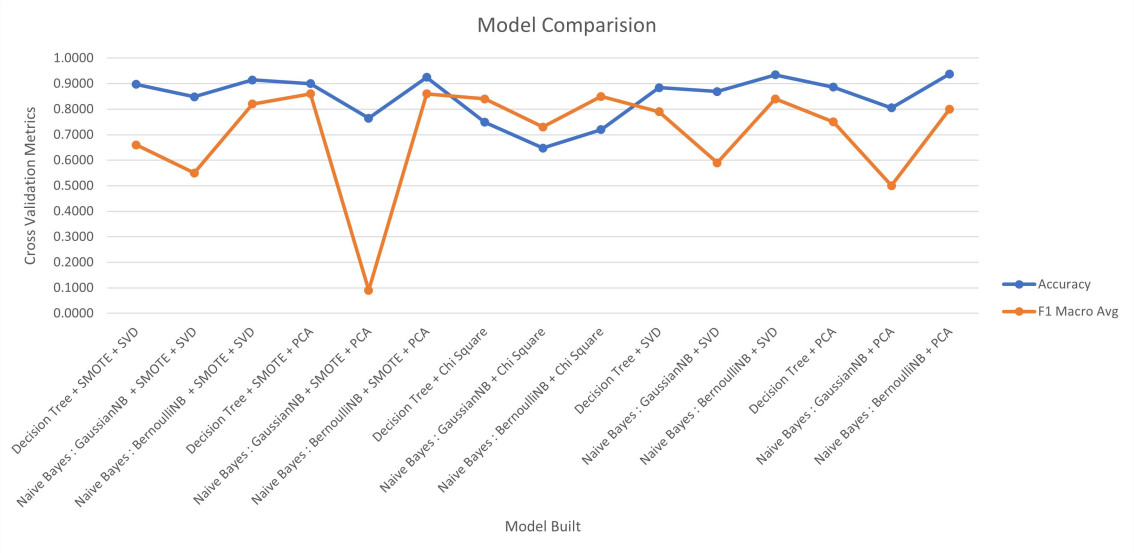

Fig 2 – Model Comparison

F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

We used the F1 score and accuracy and determined that the Bernoulli Naive Bayes classifier with Truncated SVD has provided the best results.

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`