Homework 4 CS631

Parul Kapoor 18111411 Sristi Jaiswal 18111074 Kamble Akshay Bhimrao 150317

6 October 2019

1 Problem Statement

We are given a dataset generated using the method described in the paper *"Methodology for a Security/Dependability Adaptive Protection Scheme Based on Data Mining"* by Bernabeu, Thorp, Centeno. Our aim is to create two models to distinguish between stressed grid condition and normal grid condition. The approaches tried by us are described in the following section. We used Python along with scikit-learn library, Pandas and numpy for various functions.

2 Approaches

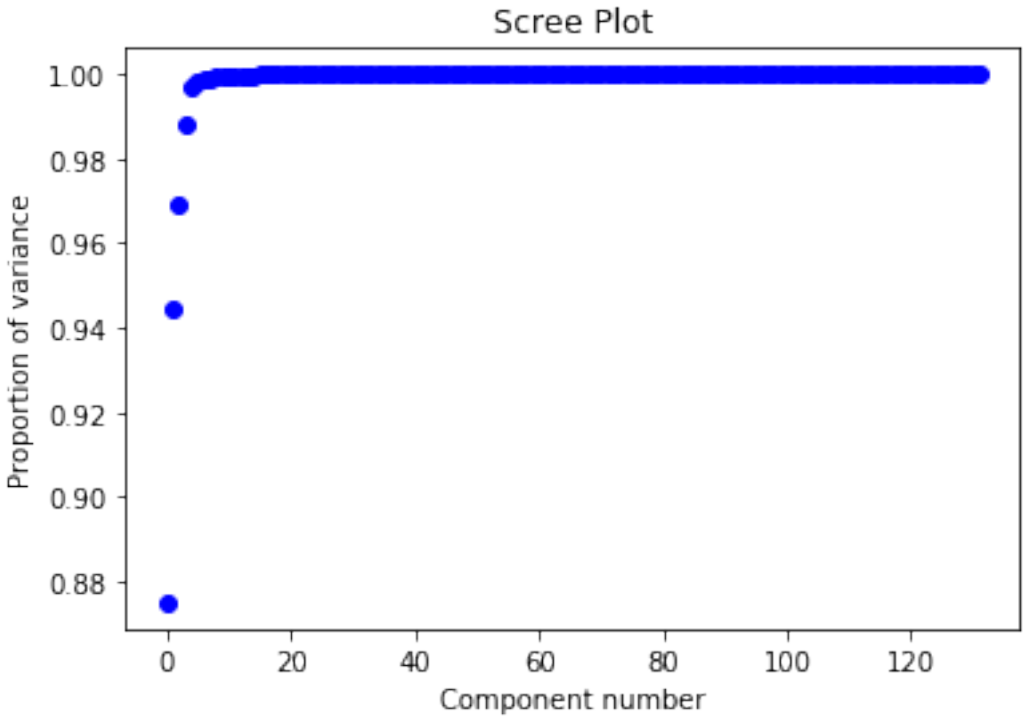
We divided our data into train(75%) and test(25%) split by randomly shuffling the data. For validation purpose, we did K-Fold cross validation using $K = 3$. This essentially takes 25% of total data for validation.

2.1 Decision Tree Classifier without PCA

We used Decision Tree Classifier from scikit learn, python and trained the model on the 75% training data and computed the accuracy, true positive, true negative, false positive, false negative and F1 score on the 25% test data as stated below-
Test Accuracy: 0.995
True positive: 627
True negative: 406
False positive: 1
False negative: 4
Test F1 score: 0.996

2.2 Dimensionality Reduction

PCA is used for linear dimensionality reduction to project the data to a lower dimensional space to make the data free of noise. We did SVD of the training data and drew a scree plot. By looking at the scree plot, we chose the number of components to be 5.



2.3 Oversampling

In order to reduce class imbalance in our given data, ie 1229 data examples for negative class whereas 1883 data examples for positive class, we have done oversampling of our negative class. We have used oversampling for the neural network.

2.4 Validation

We tried to find the best model and the best parameters of each model with the help of cross validation. The models tried are as follows-

1. **Decision Tree Classifier with PCA**:
   We transformed our data according to the principal components and finally trained a decision tree classifier on that. We tried changing various parameters like maximum height, minimum samples to split a node, etc to find the best parameters giving the minimum F1 score.

2. **Random forests with PCA**:
   We reduced the dimensionality of the data using PCA and trained an ensemble of 350 decision trees like before using scikit learn.

3. **Suport Vector Machine with PCA**:
   In this model, first we reduced the dimensionality of the data using PCA and then trained an SVM on this low dimensional data. We used K-fold cross validation to tune the parameters of SVM like kernel and class weight. We tried various kernels like 'linear', 'polynomial', 'rbf' and 'sigmoid'. Out of all, 'linear' kernel gave the best validation accuracy. Also, we have adjusted the class weight according to the ratio of the data in each class.

4. **Multilayer Perceptron with PCA**:
   We used a 1 hidden layer MLP with 5 nodes in the hidden layer on the data which is of reduced dimensionality.

5. **MLP with PCA & random oversampling**:
   Random Oversampling is just repeating of the class examples which are lesser in number. After oversampling, we have trained an MLP for binary classification. We used 1 hidden layer with 10 nodes in the hidden layer.

3 Results

Model	F1 Score	Accuracy
Decision Tree with PCA	0.986	0.984
SVM	0.991	0.990
Random Forests	0.985	0.982
Neural Network	0.996	0.997
Neural Network with oversampling	0.997	0.997

4 Selection of the best model and testing

From the results of cross validation on different models, we selected neural networks as the best model as it gave the highest F1 score. We trained the neural network on the entire oversampled training data and saved the model.
We tested the saved model on the 25% test data and reported the accuracy, true positive, true negative, false positive, false negative and the F1 score which are reported as under:
Test Accuracy: 0.997
True positive: 630
True negative: 405
False positive: 2
False negative: 1
Test F1 score: 0.998