Naive Bayes and PCA Report

Manasvi Sagarkar- 20CS60R28
Shreyas More - 20CS60R34

November 2020

1 Introduction

In this assignment we had to write a Naive-Bayes classifier. Further, we had to perform PCA and visualize the separation of data and perform feature selection using Sequential Backward selection.

2 Data Preprocessing

1. Missing data : We removed the data samples where a feature value was missing.
2. Label Encoding:
   We encoded our categorical variables such as gender,ever married, profession etc using label encoding, that is we assigned each category a numerical label. We also tried one hot encoding but did not see any performance gains from it.
3. ID column:
   We dropped the column ID because it did not help in predicting the segmentation.

3 Naive Bayes Classifier

This is the code for Naive Bayes classifier which consists of two main functions namely fit and predict. The fit method is used to train the model and the predict function is used to predict the class label of the test data. There are also two helper functions namely _predict() and _pdf() to help with the prediction. We calculate the mean, variance and prior in the fit function and calculate the posterior probabilities in the predict function. It is uses Gaussian distribution to model the class probabilty and return the class with the highest posterior probability as the predicted output.
We use k fold cross validation to ensure the model is less biased and all data points get to be a part of train and test set. We make us of stratified form

of cross validation. In K fold stratified cross validation, the splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value. The code below shows how k-fold stratified cross validation is performed where we set k = 5. The model is trained with the dataset and the final accuracy is the average across all runs.
The accuracy of the model on the test data after performing 5 fold cross validation is 48.7%.

4 PCA

For PCA, we first scaled the features such that they have the properties of a standard normal distribution, that is- they have mean 0 and standard deviation 1. We chose our PCA model to retain 95 percent of the variance and fit our model on the training data and then transformed both the training and test data. Our input data has 9 features (after deleting id) and after PCA, we end up with 8 principle components. The first two components explain 40 percent of the variance. Due to the difficulty in visualizing all 8 PCA components, we visualised the separation of data by the first two components. The data is not well seperated when projected into two dimensions which is expected since the number of principle components retained by PCA is 8. We finally used just these principle components as our features and our model accuracy after 5 fold cross validation was **49.8 percent** which is comparatively better than accuracy on full feature set.

5 Sequential backward selection

We remove outliers from the dataset by removing samples which contain even a single outlier where outlier are those with feature value > mean + 3*standard deviation. In order to do this we first convert the pandas dataframe into a numpy array and then again convert it back to a pandas dataframe after removing samples containing outlier. The reason behind removing samples with even a single outlier is otherwise the result of this operation would not be profound and the dataset would remain the same. **This leads to 163 samples or outliers being removed from the dataset.**
We perform feature selection using sequential backward selection. The crux of the idea is we begin with full feature set and calculate the accuracy of the model by dropping individual features. If there is any improvement in the accuracy of the model then the new feature set contains the original feature seat excluding the feature which led to maximum improvement in the accuracy on dropping it. The above procedure is done until there is no improvement in the accuracy of model on dropping a feature. We finally print the feature set and the model accuracy after performing 5 fold cross validation. The code to perform sequential backward selection is as follows:

After performing feature selection using sequential backward selection, the accuracy of the model on the test data is 50.04% which is comparatively better than the 48.7% accuracy obtained on using the full feature set.