# PATTERN RECOGNITION AND MACHINE LEARNING

**Course Project Report**

***Glass vs No-Glass Classification***

## 1. Importing Dependencies

Importing necessary dependencies like Pandas, NumPy, Matplotlib, Seaborn etc for data manipulation and visualization.

## 2. Loading Dataset

Mounting the Google Drive and loading the dataset in the notebook. Displaying the top 5 rows of the dataset using head() function. The dataset contains 512 latent vectors which determines whether a person is wearing glasses or not.

## 3. Concise Summary and Identification of Data Types

Using info() function to get a concise summary of the dataset which includes range index, colmuns, data types etc. Along with this, using dtype function which returns a series with data type of each column.
The dataset contains 514 columns : {'id', 'v1', 'v2', ... 'v512', 'glasses'} and each column has 4500 entries. All the entries in latent vector {'v1', 'v2', ... , 'v512'} columns are of data type float64 and the entries in {'id', 'glasses'} columns are of data type int64.

## 4. Shape of Dataset

The shape of the data set is (4500,514) because there are 514 columns and each column has 4500 entries. We get this information using shape function.

## 5. Statistical Summary of Numeric Variables

Statistical summary of the dataset is obtained using describe() function. This includes mean, standard deviation, count, minimum and maximum value etc.

## 6. Finding Null Values

To check whether the dataset contains null values or not, one can use isna() function. Here, there are no null values in any of the columns.

## 7. Visualizing the Distribution of Target Variable

To visualize the distribution of target variable {'glasses'}, we will plot a countplot using countplot() function. The countplot will give the count of observations i.e. 1 and 0.

The plot is shown below:

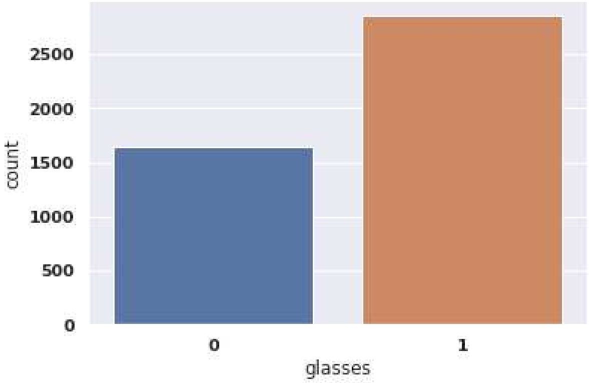

Fig. 1. Distribution of Target Variable

## 8. Seperating Target and Feature Variables and Splitting them for Training and Testing

Now, seperating the Target (y) {'glasses'} and Feature (x) {'v1', 'v2', ... , 'v512'} variables and splitting them into training and testing in the ratio 8:2 using train_test_split() function.
The shape of the x is (4500,512).

## 9. Visualizing the Latent Space

Now, we can see that x is a high dimensional data. So, to visualize it, we will be using t- distributed Stochastic Neighbor Embedding (t-SNE). It is is a tool to visualize high-dimensional data. On applying t-SNE on x, we will get an embedded data, whose shape will be (4500,2). This embedded data can be easily visualized using scatterplot() function.
The plot is given below:

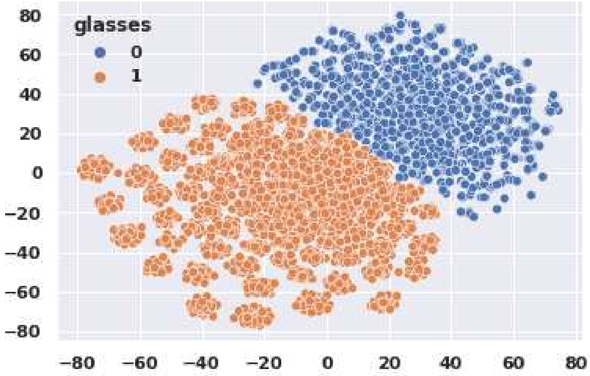

Fig. 2. Latent Space

## 10. Dimensionality Reduction

### 10.1. Principle Component Analysis (PCA)

Firstly, we will scale the features(x) using StandardScaler() function. After that, we will apply PCA such that 90% of the variance is retained by choosing the minimum number of Principal Components. The number of components obtained are 285. This means that after transformation the shape of the features will be reduced to (4500,285), initially which was (4500,512).
Finally, we will split the PCA transformed features into training and testing set.

### 10.2. Linear Discriminant Analysis (LDA)

Firstly, we will scale the features(x) using StandardScaler() function. After that, we will apply LDA keeping the number of Linear Discriminants 2. This means that after transformation the shape of the features will be reduced to (4500,1), initially which was (4500,512). Next, we will perform 5-fold cross validation and calculate score using cross_val_score() function. The cross validation score obtained for LDA is: 0.98666667, 0.98555556, 0.98777778, 0.98555556, 0.98555556
Finally, we will split the LDA transformed features into training and testing set.

### 10.3. LDA on Features obtained by PCA

Firstly,we will apply LDA keeping the number of Linear Discriminants 180 and fit transform on the features obtained by PCA.This means that after transformation the shape of the features will be reduced to (4500,1), initially which was (4500,285).
Finally, we will split the LDA_after_PCA transformed features into training and testing set.

## 11. Visualizing for Features obtained after Dimensionality Reduction

We will visualize the features obtained by PCA, LDA and LDA after PCA by plotting a scatterplot.

### 11.1. PCA

Since, the features obtained after PCA results in a high dimensional data, so to visualize it we will apply t-SNE to get an embedded data. This embedded data can be easily visualized.

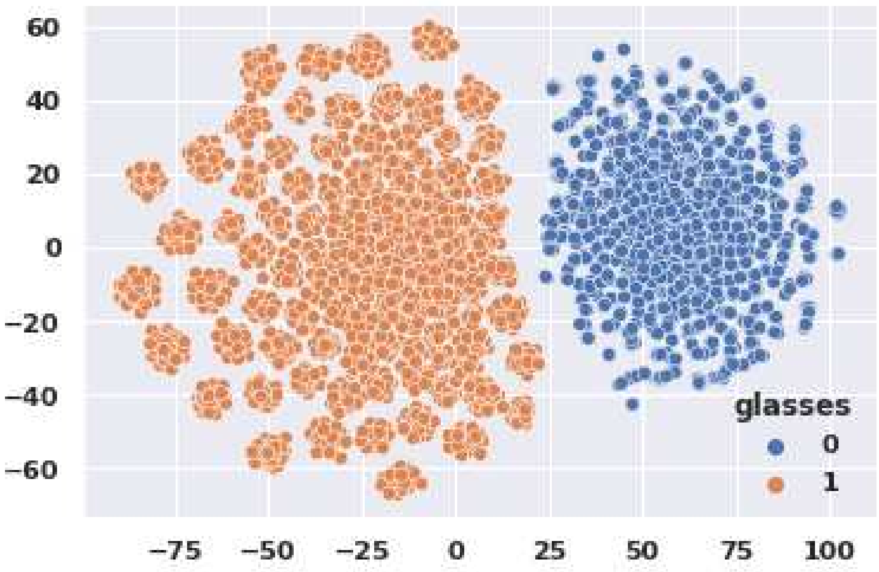

Fig. 3. Principle Component Analysis

### 11.2. LDA

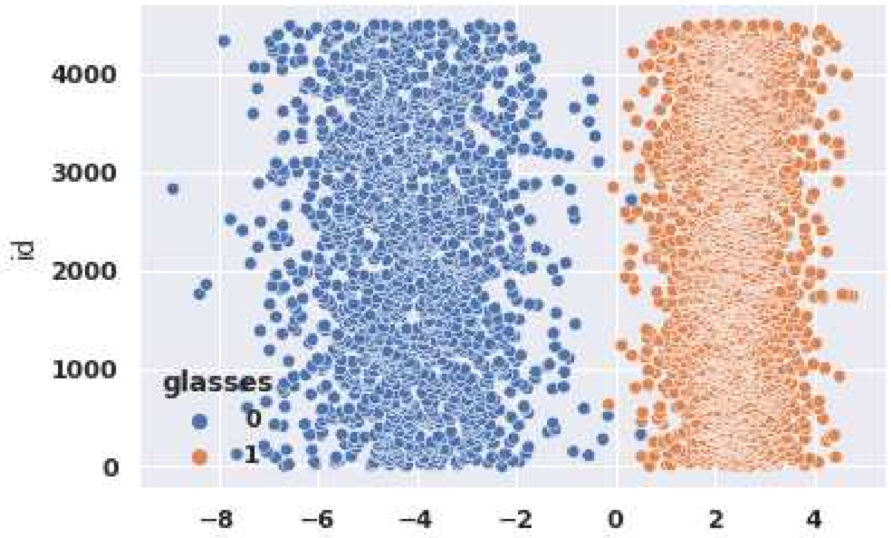

Fig. 4. Linear Discriminant Analysis

### 11.3. LDA after PCA

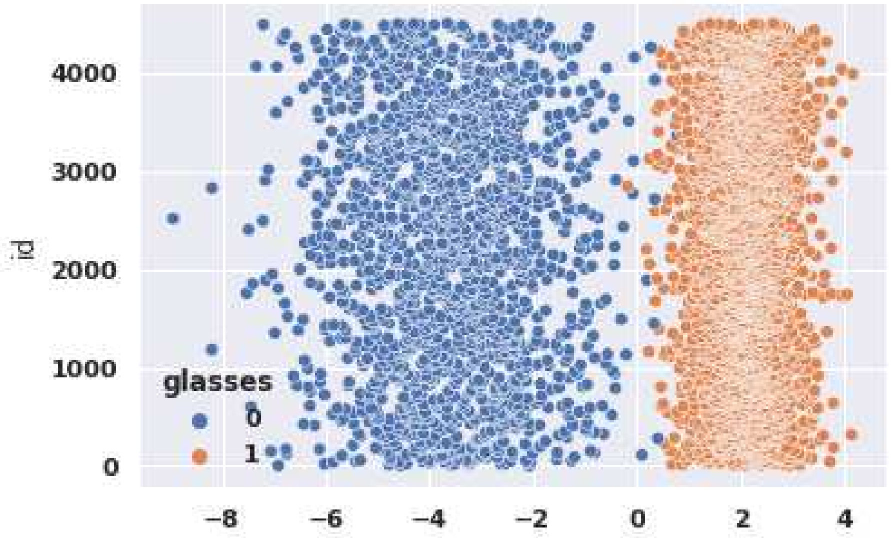

Fig. 5. LDA after PCA

## 12. Feature Selection

### 12.1. SelectKBest

Now, we will apply one of the feature selection technique, SelectKBest which select features according to the k highest score. Here, we will take k as 200. After applying, we will fit transform to get a tranformed version of x whose shape will be (4500,200), initially which was (4500,512). Finally, we will split the SelectKBest transformed features into training and testing set.

## 13. Selecting the Desired Model

Three Classifiers: Decision Tree Classifier, Multi Layer Perceptron Classifier and ADABoost Classifier.

### 13.1. Decision Tree Classifier

Firstly, we will define the classifier and tune the parameters by performing grid search.Then,we will train the Decision Tree Classifier for features with and without dimensionality reduction and feature selection and get the cross validation score for each.
On applying Grid Search, we get:

- **Best Parameters -** 'max_depth': 10, 'max_features': 0.8, 'min_samples_leaf': 20, 'min_samples_split': 0.08
- **Score -** 0.7758689813303765
- **Cross Validation Score on Best Estimator -** 0.72, 0.74555556, 0.74, 0.71888889, 0.72888889

Cross Validation Score on training with features:

- **Without Dimensionality Reduction and Feature Selection -** 0.75555556, 0.74111111, 0.75444444, 0.72555556, 0.73222222
- **With PCA -** 0.91444444, 0.94222222, 0.93000000, 0.94666667, 0.91333333
- **With LDA -** 0.99777778, 1.00000000, 1.00000000, 0.99888889, 1.00000000
- **With LDA after PCA -** 0.99888889, 0.99777778, 0.99666667, 0.99666667, 0.99777778
- **With SelectKBest -** 0.75444444, 0.74111111, 0.74444444, 0.72333333, 0.73777778

### 13.2. Multi Layer Perceptron Classifier (MLP)

We will train the Multi Layer Perceprton Classifier for features with and without dimensionality reduction and feature selection and get the cross validation score for each.
Cross Validation Score on training with features:

- **Without Dimensionality Reduction and Feature Selection -** 0.99555556, 0.99777778, 0.99555556, 0.99000000, 0.99111111
- **With PCA -** 1.00000000, 0.99777778, 0.99777778, 0.99555556, 0.99888889
- **With LDA -** 0.99777778, 1.00000000, 1.00000000, 0.99888889, 1.00000000
- **With LDA after PCA -** 0.99777778, 0.99888889, 0.99888889, 0.99666667, 0.99777778
- **With SelectKBest -** 0.99222222, 0.99555556, 0.99333333, 0.99111111, 0.98666667

### 13.3. ADABoost Classifier

We will train the ADABoost Classifier (using Decision Tree Classifier as stump) for features with and without dimensionality reduction and feature selection and get the cross validation score for each.
Cross Validation Score on training with features:

- **Without Dimensionality Reduction and Feature Selection -** 0.96777778, 0.96333333, 0.97000000, 0.97111111, 0.97666667
- **With PCA -** 1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000
- **With LDA -** 0.99777778, 0.99888889, 1.00000000, 0.99555556, 1.00000000
- **With LDA after PCA -** 0.99777778, 0.99666667, 0.99777778, 0.99555556, 0.99777778
- **With SelectKBest -** 0.96777778, 0.97000000, 0.97333333, 0.97111111, 0.97444444

## 14. Observations

- From the cross validation scores, we can see that after performing LDA after PCA, we are getting best results.
- ADABoost is performing relatively poor because it might be overfitting since its main aim is to reduce error.
- Both MLP and and Decision Tree Classifier were giving almost same results for the feature obtained from LDA after PDA.

## 15. Conclusion

We could say that MLP gives the best result without the problem of overfitting, as it was near consistent and giving best result for the feature obtained from LDA after PDA.
But Decision tree was also performing well, so if we have less computing power we can use decision tree.