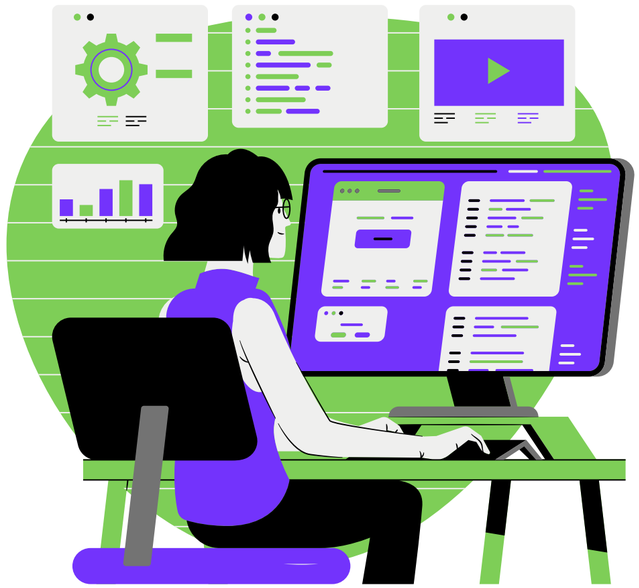
EE4708 Data Analytics Laboratory

Final Presentation

Team - Not Your Cup of C

C Gayathri (EP17B004)
N Sowmya Manojna (BE17B007)



June 1, 2021

Problem Statement

Train a model to identify devnagiri alphabet from pixel information.

1 Given Training Data:

10,000 datapoints with 10 class labels

Size of each image: 28x28 pixels

Number of Features: 784

Each image has missing pixel information that needs to be imputed

2 To do:

Train a model that can impute and predict the label for new data with high accuracy

Solution Approaches

1 Impute the Data

2 Dimensionality Reduction

3 Choose a Classifier and Train

Imputation

- Tried in-built KNN imputer (sklearn.impute.KNNImputer)
- The impued images were blurred

Own imputer

- Take a single data point and identify the 8 nearest neighbours of each missing pixel
- Assign the average of values of these neighbouring pixel to the missing feature
- Thresholded average value assignment

Dimensionality Reduction

- Tried sklearn's feature_selection.SelectKBest
- PCA with 75 components gave the best results

Choosing a Classifier

Vanilla Classifiers Tried: Bagging Classifier, Decision Tree, Random Forest, Linear Discriminant Analysis, Naive Baye's Classifiers, K Neighbours Classifier, Extra Trees Classifier, SVC with different kernels(chi2, cosine, etc.)

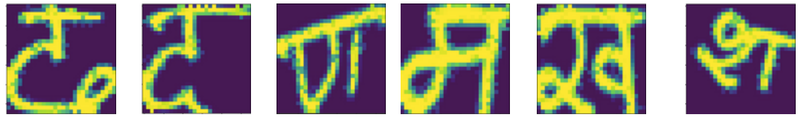
- **Best Result:** Support Vector Classifier with Radial Basis Function Kernel — test accuracy ~96.9%

Competitively Good Classifiers:

- SVC with polynomial kernel with degree = 9 and coef0 = 1 — test accuracy ~97.2% but performs poorly on public data
- K-neighbours classifier with 3 neighbours

Observations:

- SVC with RBF kernel wasn't able to differentiate well between images in the class labels 1 and 3.
- SVC with polynomial kernel was able to differentiate between 1 and 3 but struggled to distinguish 2 and 7 or 8 and 9.



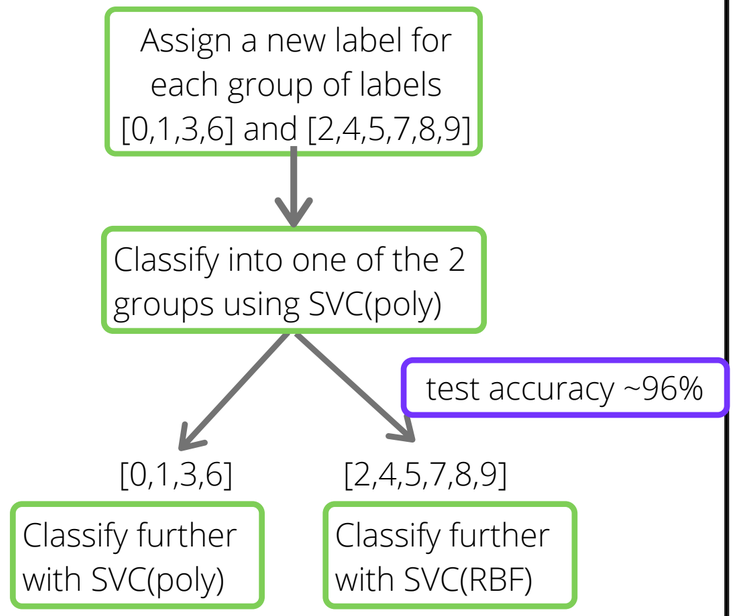
label 1 label 3 label 2 label 7 label 8 label 9

To Deal With Poor Classification of Labels 1 and 3:

- Use a vanilla SVC to classify all 10 labels
- Train an SVC on labels 1 and 3 separately
- Use this SVC on the unknown datapoints that are predicted as either 1 or 3

→ Poor results

Combining Classifiers



More Approaches

Based on Probability

Apply SVC(RBF) for all 10 classes and get the probability for each class

Check the probability of the predicted class

<0.98

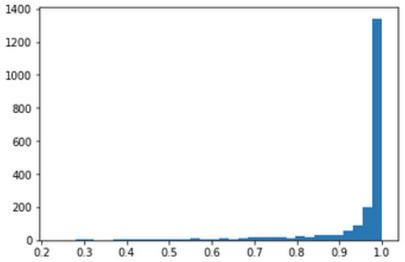
>0.98

Use a k-neighbours classifier

Use SVC(RBF)

test accuracy ~96%

Histogram of probability of predicted classes. The last bin is between 0.97747 and 1



Classifier for Each Class

Train 10 one vs all SVC(RBF)'s for each class

Predict if a datapoint belongs to a class or not when pitted against all other classes for each of 10 classes along with the corresponding probability

Count the number of classes it gets allotted to

Count:

=1

=0

>1

Allot to this single predicted class

Allot to class with maximum probability

Take every combination of classes and train 10C2 models

Use these models on every predicted pair of classes and assign the mode class

test accuracy ~94.5%

Combine with k-Neighbours

Train a k-neighbours(k=3) classifier on the 10 classes

Count the number of classes it gets allotted to

= 1

!=1

Allot to this single predicted class

Use this k-neighbours classifier and get probability for predicted class

<0.6

>=0.6

Allot to the class with maximum probability obtained from the SVC(RBF classifier)

Allot this prediction

test accuracy ~97.2%

But didn't perform well on public data

More Approaches

Other Preprocessing methods

HOG (Histogram of oriented gradients)

This method basically enables us to identify gradient orientation in localized portions of an image.

This method is relatively immune to translation, view variation of the image provided and noise too.

However, we weren't able to obtain the perfect set of parameters required for this method. Hence, the accuracies that we obtained were relatively low than when compared to our last best approach..

Image Augmentation

Affine Transformation

Image augmentation of the current train dataset was done and these images were rotated between an angle between 10 to -10. The scale of the image was also changed between 0.8 and 1.2.

This was done because most of the images that were wrongly classified largely comprised of rotated or scaled versions of the actual image.

As we had only attempted a modification for each of the train images and as we did not restrict the number of augmented images, we did not get good accuracy using this method

Combined Classifiers

We also tried a combination of classifiers - bagging and voting classifiers.

We used a combination of the following classifiers using varied parameters - K-Nearest Neighbors, Support Vector Machines and Random Forests

Although this algorithm performed almost on par with that of our last best classifier, the accuracies were slightly lesser and ence, we decided to stick to out last best model.

Final Method

Imputation
- Fill in missing value with average of 8 nearest pixel values

Splitting Data
- Use stratified shuffle split
- Ensure that the proportion of samples in each class is preserved

Principal Component Analysis
- Number of components = 75
- Performed on both test and train data

Training
- Train SVC on split train data
- Use an RBF kernel with
  - gamma = 1/(no. of features*variance)
  - regularisation parameter, C = 15
  - the penalty is a squared l2 penalty

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

Predicting
- Use this trained model on test data to predict f1 score
- Use the model on new data after PCA to predict the labels

Test Accuracy – 96.9%

References

- Kernel Functions for Machine Learning Applications
- Does it make sense to combine PCA and LDA?
- Principal Component Analysis - Explained Visually
- Face Recognition for Beginners
- Insights on Classifier Combination
- Don’t Fear the Pickle: Using pickle.dump and pickle.load
- Importance of Distance Metrics in Machine Learning Modelling
- How to Tune the K-Nearest Classifier
- What modification can improve the accuracy of an SVM algorithm?
- How can I improve the performance of SVM (Support Vector Machine)
- Support Vector Machines(SVM) — An Overview
- skimage.feature.hog module