PES UNIVERSITY	**PES UNIVERSITY** **(Established under Karnataka Act No. 16 of 2013)** **100 Ft. Road, BSK III Stage, Bengaluru – 560 085** **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** **SESSION: Jan-May 2021**

Course Title: Stochastic Models and Machine Learning	
Course code: UE20CS504	
Semester : I Sem	**Team Id: 7**
SRN: PES1PG20CS013	**Name: Krishna Desai**
SRN: PES1PG20CS001	**Name: Abdul Mueez**

ASSIGNMENT REPORT

Problem 1:

a) **Implement a Naive Bayes Classifier (NBC), of English newspaper Headlines into Politics, Sports, Education, Healthcare, Finance (5- Class Labels).**

Requirement Analysis:

The above problem aims at classifying the News Headlines into 5 Class Labels. The headlines of news should be read from a .csv file and should be classified as Headlines into Politics, Sports, Education, Healthcare, Finance (5- Class Labels).

Steps Involved:

1.Importing dataset: The main dataset consists of rows:124989 features:6 with 31 target classifier.we have applied filter conditions to get 5 Class Labels(Politics, Sports, Education, Healthcare, Finance),then applied feature selection and data cleaning over the dataset. only 2 important features we require for accurate result. final dataset named df.csv [rows:63115,features(headlines, category)].After data pre-processing, for visualization(distplot,pie-chart,countplot,wordcloud) matplotlib and seaborn libraries are used.

2. Cleaning news headline text: Removing unnecessary blank spaces and stopwords and converting data to something a computer can understand.We would not want these words to take up space in our database, or taking up valuable processing time, we can remove them easily, by storing a list of words in python) . has a list of stopwords stored in 16 different languages.

3. Word Count -CountVectorizer: Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts.

4. Term Frequency Inverse Document Frequency: Term Frequency: This summarizes how often a given word appears within a document The inverse document frequencies are calculated for each word in the vocabulary, assigning the lowest score of 1.0 to the most frequently observed word

5. Pipelining: The purpose of the pipeline is to assemble several steps that can be cross-validated together while setting different parameters.

6. Naive Bayes Classifier: Naive Bayes Classifier for Multinomial models.Sklearn already has inbuilt Multinomial Naive Bayes Classifier package Using this package we can directly train our model with the matrix obtained from Tdidf Transformer.
7. Result: Based on subject of News Headline classify it into respected category of classification.

Output Screenshots:

```
print(metrics.classification_report(y_test,predicted))
confusion_matrix=metrics.confusion_matrix(y_test,predicted)
print(confusion_matrix)
```

```
              precision    recall  f1-score   support

           0       0.72      0.14      0.23       190
           1       0.84      0.86      0.85      2877
           2       0.74      0.38      0.50       817
           3       0.79      0.62      0.69      1338
           4       0.82      0.95      0.88      6571
           5       0.88      0.62      0.73       830

    accuracy                           0.82     12623
   macro avg       0.80      0.59      0.65     12623
weighted avg       0.82      0.82      0.81     12623

[[  26   17   11   16  120    0]
 [   1 2464   12   35  340   25]
 [   1   45  310   83  369    9]
 [   2   95   38  826  363   14]
 [   4  196   40   72 6237   22]
 [   2  111   10   15  178  514]]
```

Model is giving 82% accuracy for given classification

```
print('actual_ values:',y_test)

print('predicted_values:',predicted)
```

```
actual_ values: [1 4 5 ... 1 4 3]
predicted_values: [1 4 5 ... 1 4 3]
```

Testing on New Headline:

```
#testing
t1='education is power'
print(categories[text_clf.predict([t1])])
```

```
['EDUCATION']
```

```
t2="David Cross Proves Yet Again That He's Terrible At Apologizing"
print(categories[text_clf.predict([t2])])
```

```
['POLITICS']
```

```
t3="high price crazybusy"
print(categories[text_clf.predict([t3])])
```

```
['FINANCE']
```

```
t4=df.headline[2303]  ## showing missclassification
print(t4)
print(categories[text_clf.predict([t4])])
```

```
Katy Perry Disses Taylor Swift On 'American Idol' Because Feuds Die Hard
['POLITICS']
```

Interpretation of efficiency:

```
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(x_test, y_test)
print(result)
```

```
0.822070823100689 2
```

model is giving 82.2% accuracy that means the model is successfully classifying the new News headline into respected category with the 82% accuracy.

Learning Outcome:

- Dataset cleaning and processing of text cleaning.
- How sentiment analysis is done using vectorization and tokenization.
- Applications of the Naïve Bayes algorithm.

b) For the same Dataset in (a) apply One-Versus-Rest SVM and classify.

Steps Involved:

1. import all the necessary libraries

2.Data Cleaning: Removing unnecessary urls,special characters,numbers,tabs,line jump,white space.

3.Loading universal sentences encoder:pre-trained sentence encoders by Google, ready to convert a sentence to a vector representation without any additional training, in a way that captures the semantic similarity between sentences.

4.Sentences embedding: techniques represent entire **sentences** and their semantic information as vectors. This helps the machine in understanding the context, intention, and other nuances in the entire text.

5.Divide the data into training set and testing set for modelling

6.Modelling using One-Versus-Rest SVM Support Vector Machine

SVM Classifier:

SVM classifiers do not just find a line (or in high dimensions, a hyperplane) that separates the two classes. They try to find the best line that separates them. The objective of SVM classifiers is to maximize the margin between the positive class and the negative class. This margin is defined as the distance between two Support Vectors, hence the name.

One-vs-Rest

One-vs-rest (OvR for short, also referred to as One-vs-All or OvA) is a heuristic method for using binary classification algorithms for multi-class classification. It involves splitting the multi-class dataset into multiple binary classification problems.

7.Result: Based on subject of News Headline classify it into respected category of classification.

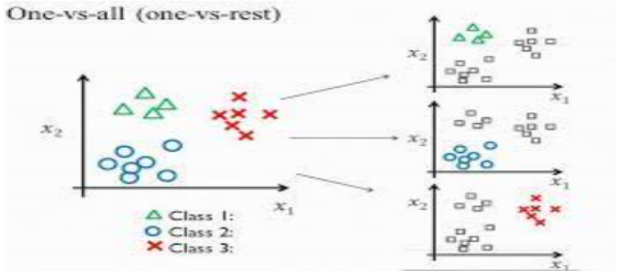


fig:One Vs Rest classification

Output Screenshots:

```
              precision    recall  f1-score   support

           0       0.78      0.49      0.60        37
           1       0.90      0.91      0.90       492
           2       0.79      0.66      0.72       155
           3       0.82      0.80      0.81       225
           4       0.91      0.95      0.93      1145
           5       0.91      0.77      0.84       155

    accuracy                           0.89      2209
   macro avg       0.85      0.77      0.80      2209
weighted avg       0.89      0.89      0.88      2209

[[  18    1    1    3   13    1]
 [   1  448    3    4   30    6]
 [   0    8  103   15   29    0]
 [   2    8    9  181   23    2]
 [   2   17   15   17 1091    3]
 [   0   17    0    0   18  120]]
```

```
accuracy = accuracy_score(test_labels,prediction)
accuracy
```

```
0.8877320054323223
```

Interpretation of efficiency:

```
accuracy = accuracy_score(test_labels,prediction)
accuracy
```

```
0.8877320054323223
```

SVM Model gives 88.77% Accurate result for new training example

Learning Outcome:

- Headlines Conversion of Vector, using Google Sentence Embedding
- Application of One-Versus-Rest SVM model

Problem 2: Implement a Multi-Layer (One Input, One Output and One or more Hidden Layers) ANN for handwritten digit classification using MNIST dataset.

Requirement Analysis:

The above problem aims at being able to recognize Handwritten Digit when the system has been priorly trained by a set of arrays to allow for feature extraction. We are required to design and use a multilayer ANN to solve this problem.

Steps Involved:

1) collect the images of handwritten digits written on white sheet.
2) Design and develop the neural network.
3) Provide the images as input for training the network.
4) Test the system with random test images to observe the accuracy score.

Artificial neural networks (ANNs)

It is simply called neural networks (NNs), are computing systems vaguely inspired by the biological neural networks that constitute animal brains.its a collection of connected units or nodes called artificial neurons
To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.Artificial Neural Network primarily consists of three layers:

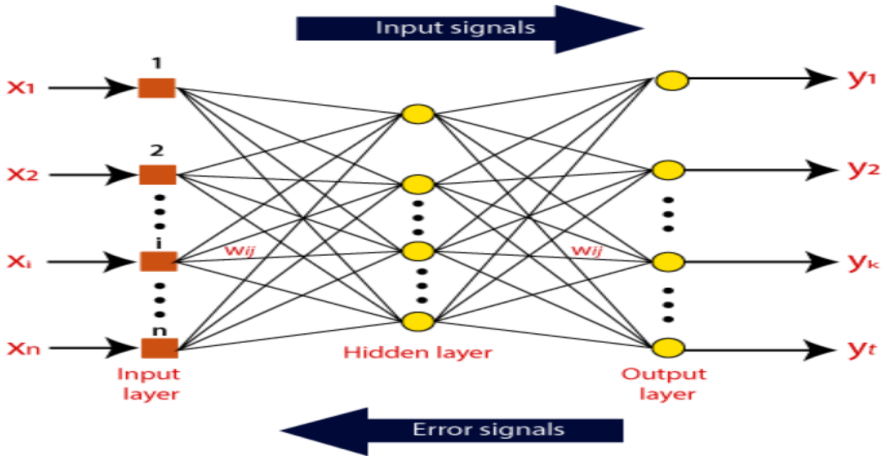


Fig1: A fully connected layer with one hidden layers

Dataset Details:

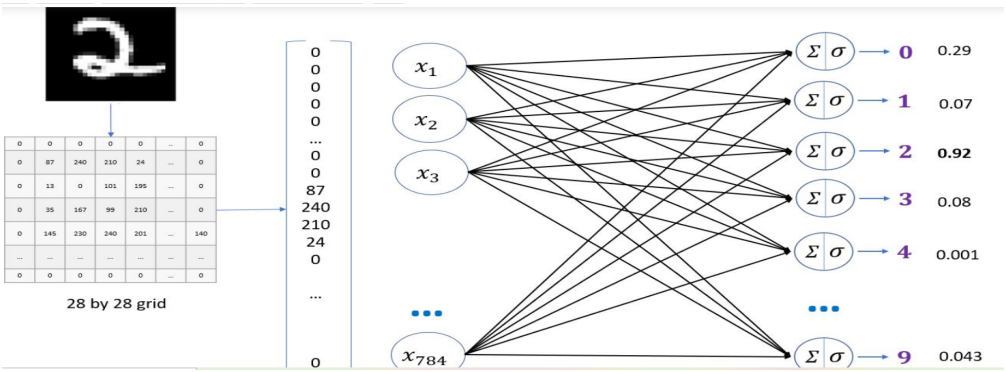
-MNIST dataset can be accessed from MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges that we have directly loaded from keras.datasets import mnist.
-Split the given dataset into a train and test set for modelling.

Steps of Implementation:

- here we will use 10 neurons because our target function is classified into 0 to 9 nominal data.
- since we are using sigmoid function output will be between 0 to 1.
- input image will be classified into one of the target labels.

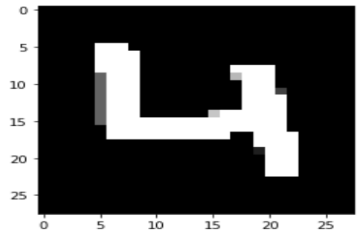
1. we supply this image to 2D array
2. 2D --> 1D flattened array

3. Each pixel will input to each neuron x1->0,x2->0....x784->0 here we have 784 features that act as the first layer of the Neural Network.
4. we can add hidden layers to improve the model accuracy
5. The output layer has 10 neurons because our target function is classified into 0 to 9 nominal data.
6. since we are using sigmoid function output will be between 0 to 1.
7. input image will classify into one of the target labels according to maximum of all output here image is predicted as digit 2 with the highest value of sigmoid function =0.92
8. Add more layers to improve the accuracy of model
9.Preprocessing the real life image



Output Screenshots 1:

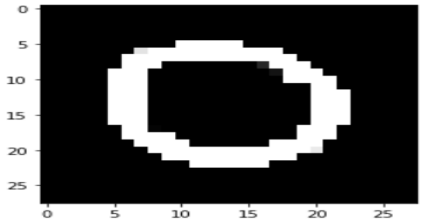
```
=========PREDICTION============
```



```
Final Output: 4
Prediction (Softmax) from the neural network:
 [[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]]
```

Output Screenshots 2:

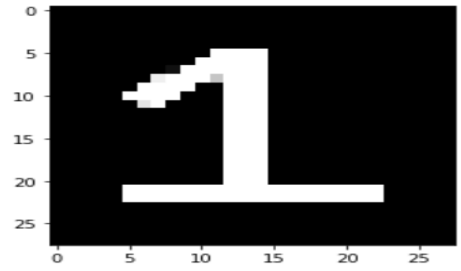
```
=========PREDICTION============
```



```
Final Output: 0
Prediction (Softmax) from the neural network:
 [[1. 0. 0. 0. 0. 0. 0. 1. 0. 1.]]
```

Output Screenshots 1:

```
=========PREDICTION============
```




```
Final Output: 1

Prediction (Softmax) from the neural network

 [[0. 1. 1. 1. 0. 1. 0. 0. 1. 0.]]
```

Interpretation of efficiency:

```
: model.evaluate(X_test_flattend,y_test)

  WARNING:tensorflow:Model was constructed with shape (None, 28, 28) for input KerasTensor(ty
  28), dtype=tf.float32, name='flatten_1_input'), name='flatten_1_input', description="create
  it was called on an input with incompatible shape (None, 784).
  313/313 [==============================] - 1s 1ms/step - loss: 0.0880 - accuracy: 0.9728

: [0.08796416968107224, 0.9728000164031982]
```

after adding layers model is giving 97.22% accuracy

Learning Outcome:

- Through the completion of this assignment we were able to learn how neural networks work internally.
- It also gave us the opportunity to understand the versatility of applications of a Neural network.
- we were also able to learn how to deal with large dataset

References:

1) Machine Learning , by Tom Mitchel
2) Tutorial to build a Handwritten Digit Recognition System. (Assignment Document)
3) Stack overflow and GitHub to help solve errors.
4) Towards data science machine learning

Name and Signature of the Faculty