HOCHSCHULE
SRH HEIDELBERG
SRH University Heidelberg

Task 1
Handwritten Digit Classification

Name:	Shrey Kothavade
Matriculation Number:	11018044
Email Address:	11018044@stud.hochschule-heidelberg.de
Supervisor:	Prof. Dr.-Ing. Milan Gnjatovic
Project Begin Date:	….
Project End Date:	9th November 2022

Abstract:

The recognition of handwritten digits has been approached as a multi-class classification issue in the perspective of machine learning, where digits (0-9) are seen as a class, and the machine learning task is primarily to train a classifier that can competently discriminate the classes. The accuracy of a single classifier trained using a conventional learning algorithm typically varies across various datasets, indicating that the same learning method may generate strong classifiers on some datasets while weak classifiers on other datasets. In this report, we used MNIST dataset, consisting of data derived from 70000 handwritten number images in CSV format. The classification of the numbers is based on gaussian naïve bayes approach.

I. Introduction:

A subcategory of artificial intelligence is machine learning. Without being directly programmed to do so, machine learning algorithms generate a model using sample data, sometimes referred to as training data, so predictions or judgments can be made. Machine learning algorithms are applied in a broad range of industries where it is challenging to create traditional algorithms to carry out the required functions, including computer vision, voice recognition, digit recognition, and email filtering.

In this report, we are mainly focusing on implementation of Gaussian naïve bayes approach on the dataset from MNIST (Modified National Institute of Standards and Technology database) for Handwritten Digit Classification.

II. Naïve-Bayes approach

Based on the Bayes theorem, the probabilistic machine learning algorithm known as Naive Bayes is utilized for numerous classification applications. Naïve Bayes is the extension of naïve Bayes. The Gaussian or normal distribution is the most straightforward to implement among the several functions used to estimate data distribution since you need to determine the mean and standard deviation for the training data.

A probabilistic machine learning algorithm called Naive Bayes can be applied to a variety of classification applications. Naive Bayes is frequently used for document categorization, spam filtering, prediction, and other tasks. This algorithm takes its name from Thomas Bayes' findings, on which it is based.

Bayes Rule = P (Y | X) = P (X | Y) * P (Y) / P (X)

If we assume that X’s follow a Gaussian or normal distribution, we must substitute the probability density of the normal distribution and name it Gaussian Naïve Bayes. To compute this formula, you need the mean and variance of X.

$$P(X|Y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

In the above formulae, sigma and mu is the variance and mean of the continuous variable X computed for a given class c of Y.

III. Naïve Bayes Classifier:

The Nave Bayes classifier assumes that the values of each feature are independent of one another. To estimate the classification-related parameters, naive Bayes classifiers need training data. Naive Bayes classifiers may be used in a variety of real-world situations due to their easy design and implementation.

IV. Pre-processing dataset:

The original dataset of handwritten digits is in image format. These images have been divided in 28*28 pixel format. The value if each pixel is then stored in a CSV format file. Where the format of the data generated is "label, pix-11, pix-12, pix-13, ...". In this Label stands for the value of the digit, and "pix-ij" is the pixel in the i^{th} row and j^{th} column. The MNIST dataset is generated from 70000 images and divided in 60000 train set images and 10000 test set images. As stated earlier, the dataset is in CSV format, i.e., the images pixels values are stored in it.

V. Training and testing procedures

The data imported from MNIST database is divided prior training and testing.

The data is trained with Gaussian Naïve Bayes approach. For this, library from SKLearn has been used. "from sklearn.naive_bayes import GaussianNB" has been used to import the gaussian library. This function is then added in a variable and ".fit" command is used to train the model. This trained model is then used to predict the test set. From which we can get the Accuracy of the trained model, classification report and confusion matrix. From the trained model, if we want to predict a single image, multiple commands are being introduced at the end of the code. The condition here is the data should be in array or CSV format to give it to the model.

VI. Challenges faced

1. While working, the dataset from given website was not loading and training. With the commands of "train_set=pd.read_csv('[#location_of_the_folder]')" I was able to load the data, but the pre processing of the data faced multiple issues, as the data didn't had any headers. When I tried to add the headers manually, the file read used to be in a single column. As it was happening repeatedly and the issue was difficult to solve, I preferred to fetch dataset direct online.
2. After loading the data directly from online, the reshaping of the data was important. I faced issues in understanding the syntax at start.
3. The prediction of a number from a single image data was a difficult task to figure out how to execute. As python was new to me.

VII. Results

The quality of any model is based on Accuracy, Precision, Recall and F1 score.

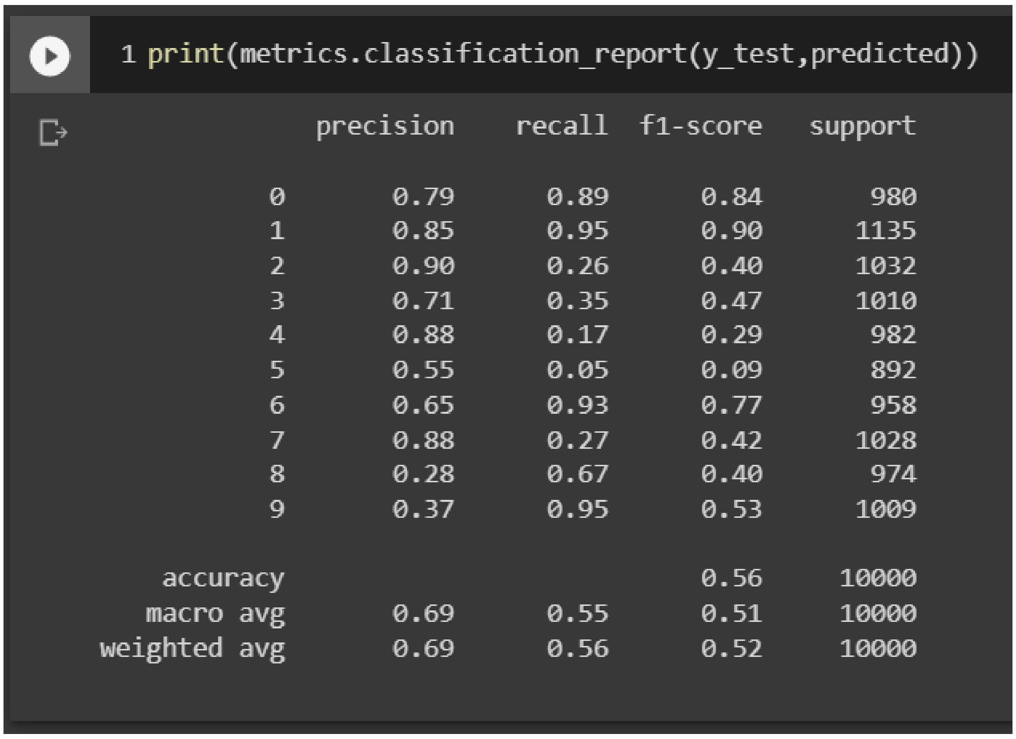
Accuracy: One metric for measuring classification model performance is accuracy. Simply In layman's term, accuracy is the fraction of predictions that our model correctly predicted. Accuracy is defined as follows in formal language: Accuracy is the quantity of accurate forecasts by sum of all projections.

Precision: Precision, or the quality of a successful prediction produced by the model, is one measure of the ML model's performance. Precision is calculated by dividing the total number of positive predictions by the proportion of true positives (i.e., the number of true positives plus the number of false positives).

Recall: Recall is a statistic that measures the proportion of accurate positive predictions among all possible positive predictions. The recall is determined as the proportion of Positive samples that were properly identified as Positive to all Positive samples. The recall assesses how well the model can identify positive samples. The more positive samples that are identified, the higher the recall.

F1 Score: One of the most crucial assessment measures in machine learning is the F1-score. By combining precision and recall, two measures that would normally be in competition, it elegantly summarizes the prediction ability of a model.

Our model got the accuracy of 55.58%. Precision, Recall and F1 score for individual classes are as follows:



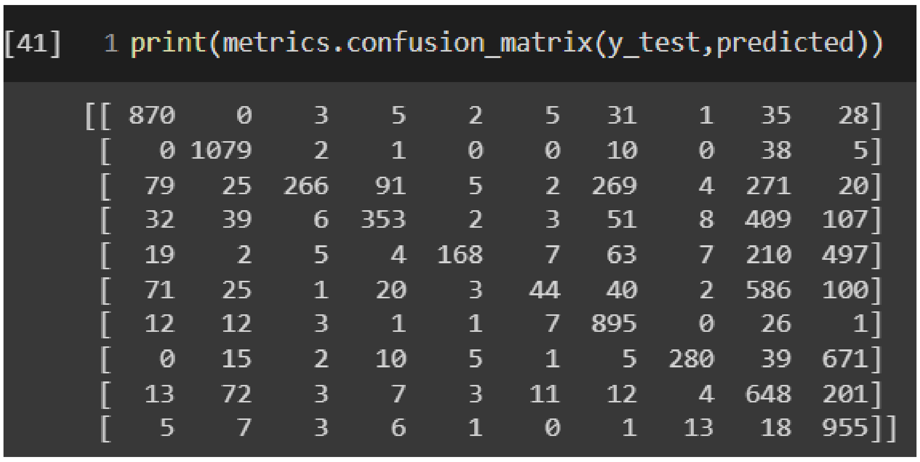
```
print(metrics.classification_report(y_test,predicted))
```

```
              precision    recall  f1-score   support

           0       0.79      0.89      0.84       980
           1       0.85      0.95      0.90      1135
           2       0.90      0.26      0.40      1032
           3       0.71      0.35      0.47      1010
           4       0.88      0.17      0.29       982
           5       0.55      0.05      0.09       892
           6       0.65      0.93      0.77       958
           7       0.88      0.27      0.42      1028
           8       0.28      0.67      0.40       974
           9       0.37      0.95      0.53      1009

    accuracy                           0.56     10000
   macro avg       0.69      0.55      0.51     10000
weighted avg       0.69      0.56      0.52     10000
```

The confusion Matrix for the classification is:



```
[41]  print(metrics.confusion_matrix(y_test,predicted))

    [[ 870    0    3    5    2    5   31    1   35   28]
     [   0 1079    2    1    0    0   10    0   38    5]
     [  79   25  266   91    5    2  269    4  271   20]
     [  32   39    6  353    2    3   51    8  409  107]
     [  19    2    5    4  168    7   63    7  210  497]
     [  71   25    1   20    3   44   40    2  586  100]
     [  12   12    3    1    1    7  895    0   26    1]
     [   0   15    2   10    5    1    5  280   39  671]
     [  13   72    3    7    3   11   12    4  648  201]
     [   5    7    3    6    1    0    1   13   18  955]]
```

VIII. Questions addressed in task:

1. How do you handle features with a constant value across all images in the dataset?

- Use the “get_constant_features” functions to get all the constant features.
- Store all the constant features as a list for removing from the dataset.
- Drop all such features from the dataset.

2. Do you, and if so, how do you the prevent arithmetic underflow or overflow?

Underflow happens when numbers near to zero are rounded to zero, whereas Overflow happens when we have completed a computation but the result is bigger than the greatest number we can represent. The underflow occurs when the product of the probabilities we obtain is too little to be captured by our machine-learning model. To avoid this, log probabilities are used. The alternative methods are:

- testing based on valid input ranges
- validation using formal methods
- use of invariants

3. How do you explain the low classification accuracy obtained by applying the Gaussian naive Bayes approach in this particular context?

The assumption that all features are independent is not usually the case in real life so it makes naive bayes algorithm less accurate than complicated algorithms.

IX. Conclusion:

Handwritten Digit Classification has been successfully implemented with the Gaussian Naïve Bayes Approach on the Google colab platform using Python language. Various libraries are imported for the execution of the model. The accuracy of the model is 55.58%. It was observed that the model faced issues in detecting number 8 and 9 more.

X. References:

https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed
https://www.baeldung.com/cs/naive-bayes-classification-performance
https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/?
https://stackoverflow.com/questions/40535925/sklearn-naive-bayes-classifier-gives-low-accuracy
https://towardsdatascience.com/essential-things-you-need-to-know-about-f1-score-dbd973bf1a3#:~:text=F1%2Dscore%20is%20one%20of,competing%20metrics%20%E2%80%94%20precision%20and%20recall.
https://www.upgrad.com/blog/gaussian-naive-bayes/
https://www.upgrad.com/blog/naive-bayes-classifier/
https://www.researchgate.net/publication/228861610_Handwritten_digit_classification