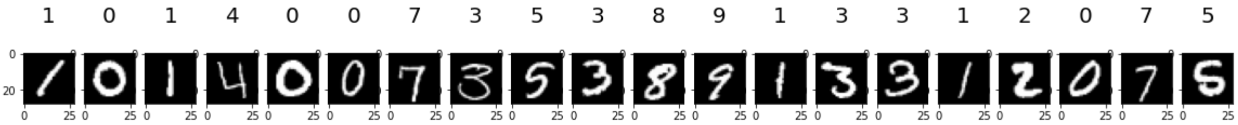
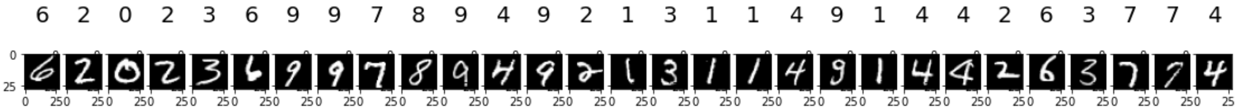
OBSERVATIONS AND FINAL REPORT

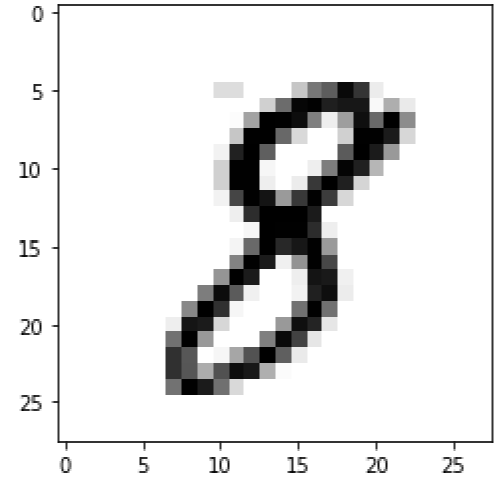
Members: Aakanksha Agrawal DIYA KHURDIYA Sammy Nyakabau

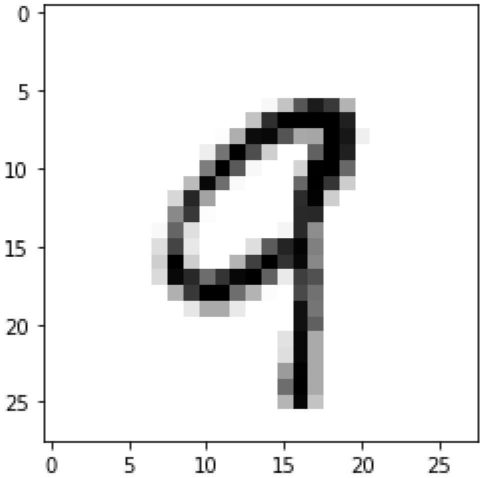
Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9)

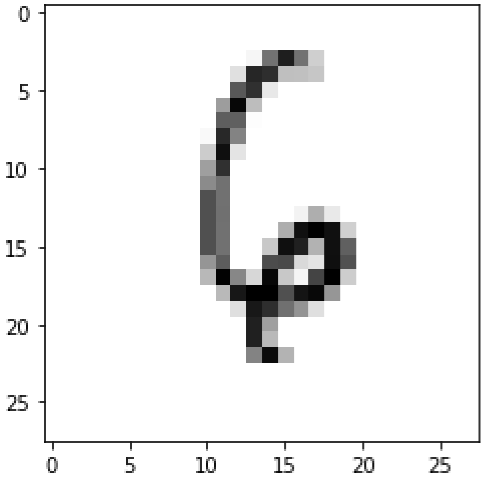
The plotting of random MNSIT dataset:

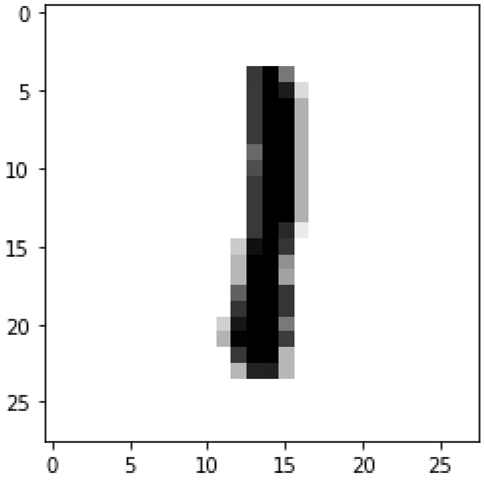




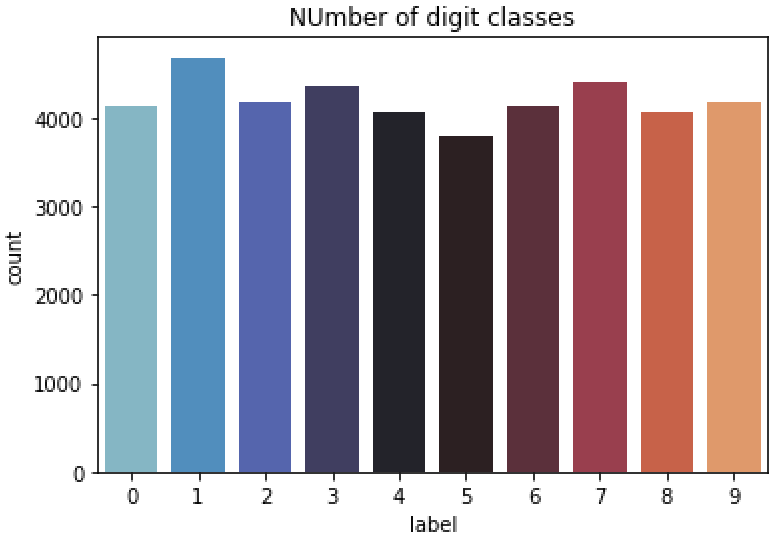








Frequency plot of input labels:

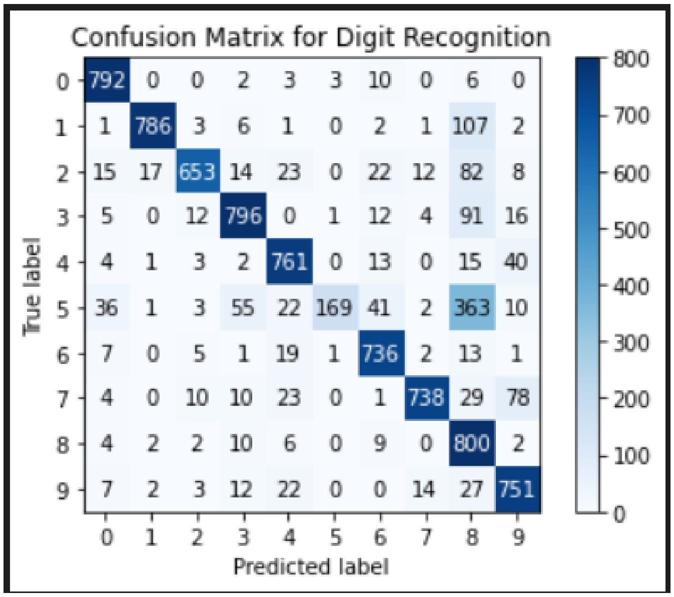


1. **Logistic Regression**
   Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data.
   Visualizations - cos and alpha graphs
   a. Accuracy: 85%
   b. Limitations:
      i. The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables

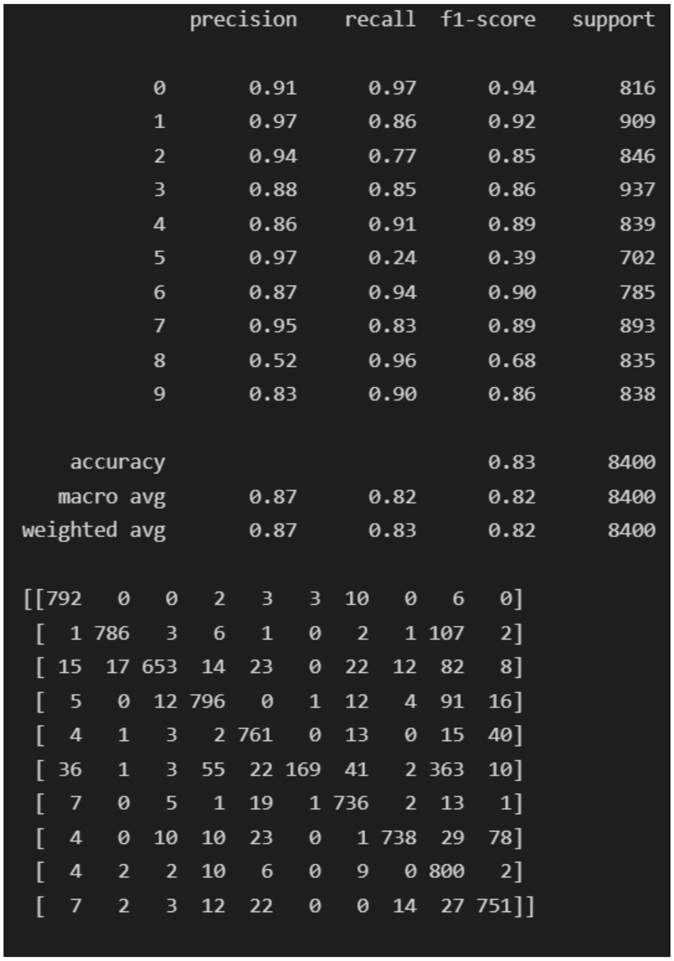
   c. Advantages:
      i. Easier to implement, train and interpret.
      ii. Easily expandable to multiple classes and provides a normalized probability distribution.
      iii. Measures of how accurate a predictor is and it’s direction of association(positive/negative)
      iv. Decent accuracy for simple datasets and performs well when data is linearly separable i.e. when data graphed in two dimensions can be separated through a straight line.

v. It can overfit in high dimensional datasets then we can use regularization technique to avoid overfitting (low training error and high testing error)

d. Confusion Matrix is a performance measurement for machine learning classification



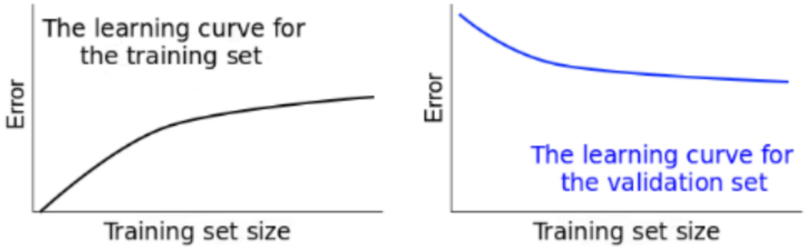
	0	1	2	3	4	5	6	7	8	9
0	792	0	0	2	3	3	10	0	6	0
1	1	786	3	6	1	0	2	1	107	2
2	15	17	653	14	23	0	22	12	82	8
3	5	0	12	796	0	1	12	4	91	16
4	4	1	3	2	761	0	13	0	15	40
5	36	1	3	55	22	169	41	2	363	10
6	7	0	5	1	19	1	736	2	13	1
7	4	0	10	10	23	0	1	738	29	78
8	4	2	2	10	6	0	9	0	800	2
9	7	2	3	12	22	0	0	14	27	751

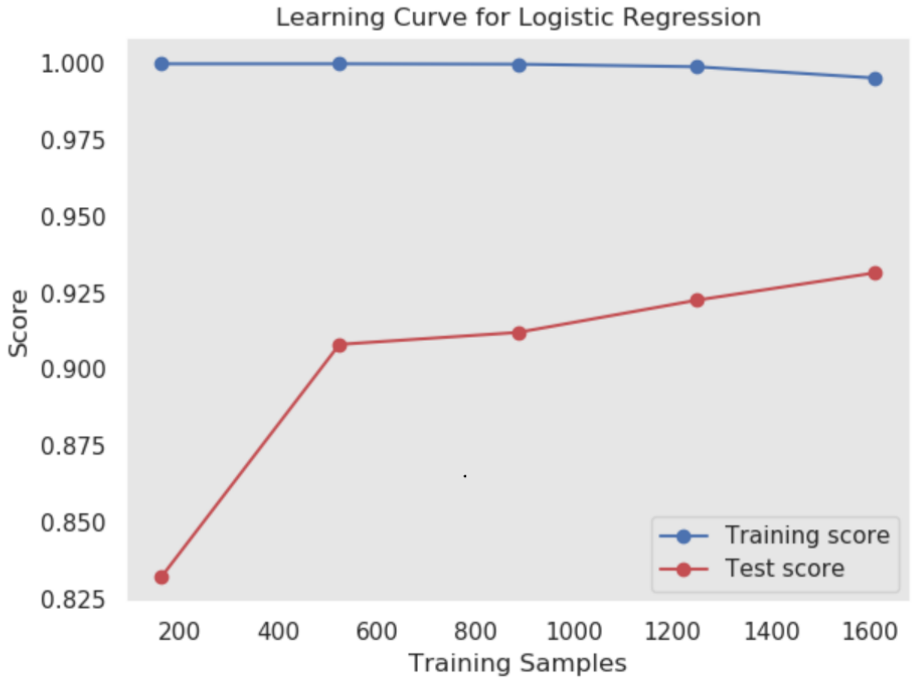


```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       816
           1       0.97      0.86      0.92       909
           2       0.94      0.77      0.85       846
           3       0.88      0.85      0.86       937
           4       0.86      0.91      0.89       839
           5       0.97      0.24      0.39       702
           6       0.87      0.94      0.90       785
           7       0.95      0.83      0.89       893
           8       0.52      0.96      0.68       835
           9       0.83      0.90      0.86       838

    accuracy                           0.83      8400
   macro avg       0.87      0.82      0.82      8400
weighted avg       0.87      0.83      0.82      8400

[[792   0   0   2   3   3  10   0   6   0]
 [  1 786   3   6   1   0   2   1 107   2]
 [ 15  17 653  14  23   0  22  12  82   8]
 [  5   0  12 796   0   1  12   4  91  16]
 [  4   1   3   2 761   0  13   0  15  40]
 [ 36   1   3  55  22 169  41   2 363  10]
 [  7   0   5   1  19   1 736   2  13   1]
 [  4   0  10  10  23   0   1 738  29  78]
 [  4   2   2  10   6   0   9   0 800   2]
 [  7   2   3  12  22   0   0  14  27 751]]
```

Misclassification might occur if the model is too simple or the data is very noisy (data not so stable with high variability). The multiple classes might not be linearly separable. The problem of overfitting (model fits exactly against its training data but algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose) might be another factor.

Accuracy and performance can be used by better scaling and pre-processing techniques. Using a different set of optimal values for the hyperparameters such that it tunes perfectly. Increasing the training data sample can also contribute to increasing the accuracy.





2. SVM

Given labeled training data the algorithm outputs the best hyperplane which classified new examples. In two-dimensional space, this hyperplane is a line splitting a plane into two parts where each class lies on either side. The intention of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that separately classifies the data points.

a. Accuracy: 81%

b. Limitations:

i. Long training time for large datasets

ii. Since the final model is not so easy to see, we can not do small calibrations to the model hence its tough to incorporate our business logic

c. Advantages:

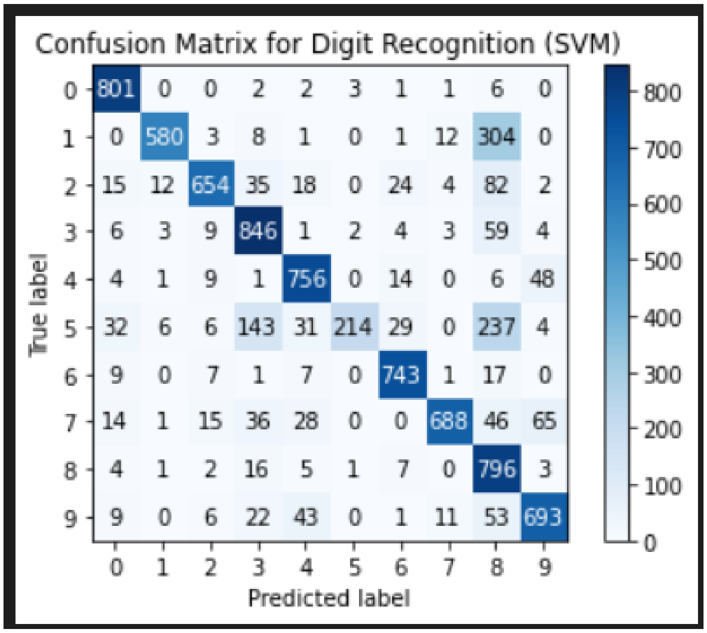
i. Memory efficient

ii. SVM works relatively well when there is a clear margin of separation between classes

iii. more effective in high dimensional spaces

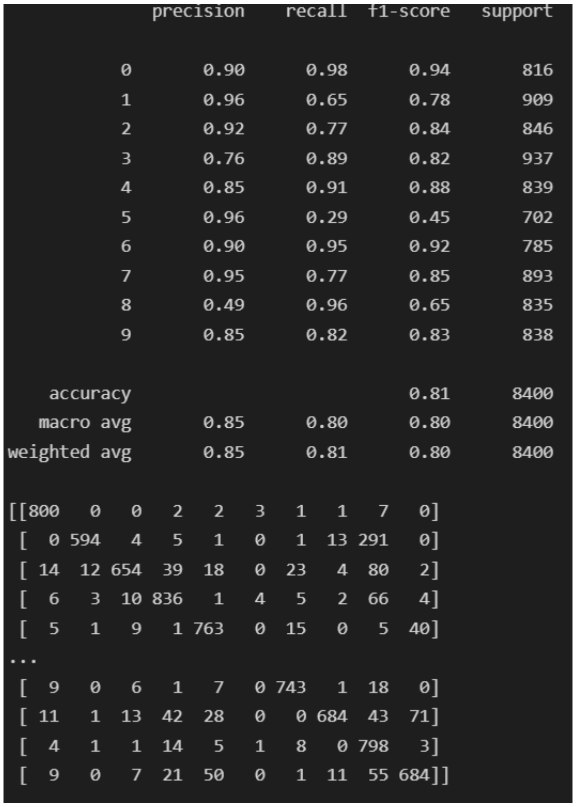
iv. Works well with even unstructured and semi structured data like text and images

v. There are many reasons to prefer each type of classifier over others in different circumstances (e.g. time/memory required for training/evaluation, amount of tweaking/exploration required to get a decent working model, etc.). There are certainly domain-specific tricks than can make classifiers more suitable for digit recognition. Some of these tricks work by increasing invariance to particular transformations that one would expect in handwritten digits (e.g. translation, rotation, scaling, deformation).

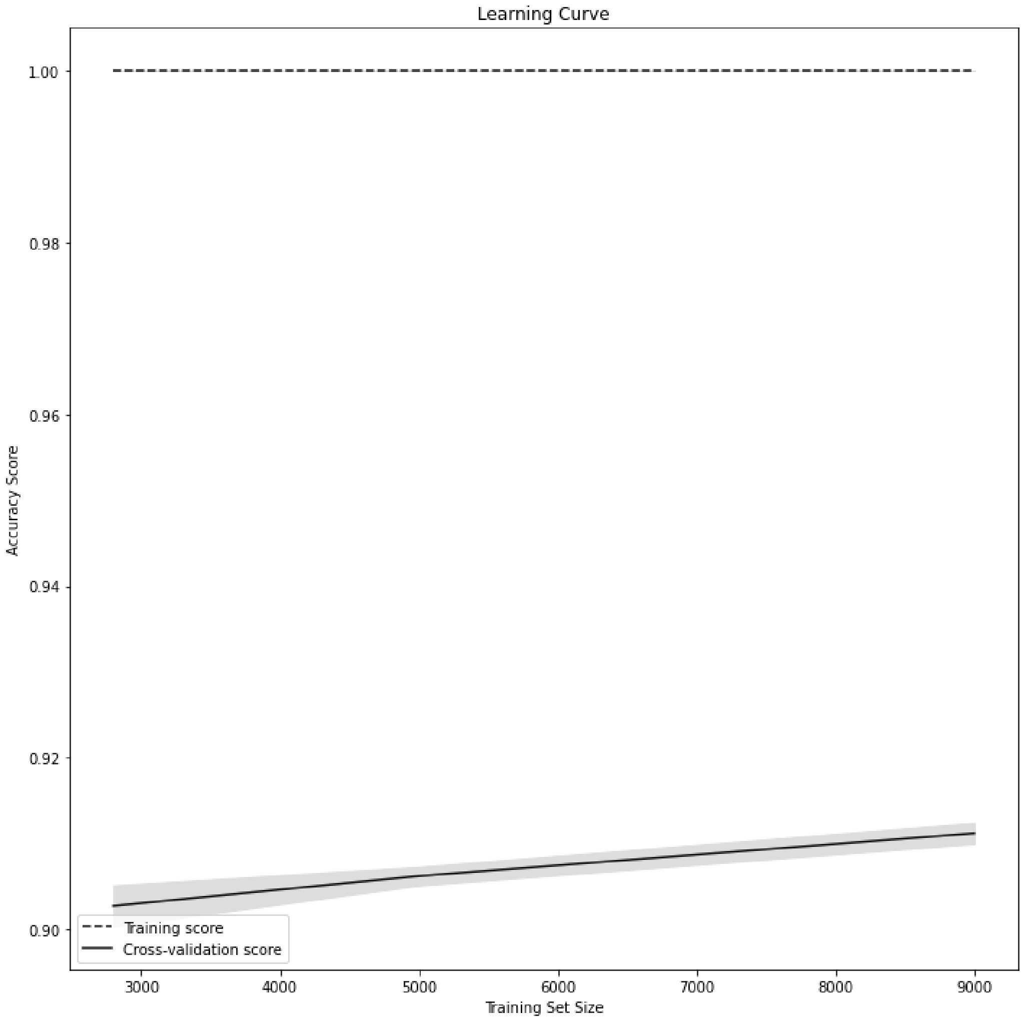


Confusion Matrix for Digit Recognition (SVM)

	0	1	2	3	4	5	6	7	8	9
0	801	0	0	2	2	3	1	1	6	0
1	0	580	3	8	1	0	1	12	304	0
2	15	12	654	35	18	0	24	4	82	2
3	6	3	9	846	1	2	4	3	59	4
4	4	1	9	1	756	0	14	0	6	48
5	32	6	6	143	31	214	29	0	237	4
6	9	0	7	1	7	0	743	1	17	0
7	14	1	15	36	28	0	0	688	46	65
8	4	1	2	16	5	1	7	0	796	3
9	9	0	6	22	43	0	1	11	53	693



```
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       816
           1       0.96      0.65      0.78       909
           2       0.92      0.77      0.84       846
           3       0.76      0.89      0.82       937
           4       0.85      0.91      0.88       839
           5       0.96      0.29      0.45       702
           6       0.90      0.95      0.92       785
           7       0.95      0.77      0.85       893
           8       0.49      0.96      0.65       835
           9       0.85      0.82      0.83       838

    accuracy                           0.81      8400
   macro avg       0.85      0.80      0.80      8400
weighted avg       0.85      0.81      0.80      8400

[[800   0   0   2   2   3   1   1   7   0]
 [  0 594   4   5   1   0   1  13 291   0]
 [ 14  12 654  39  18   0  23   4  80   2]
 [  6   3  10 836   1   4   5   2  66   4]
 [  5   1   9   1 763   0  15   0   5  40]
...
 [  9   0   6   1   7   0 743   1  18   0]
 [ 11   1  13  42  28   0   0 684  43  71]
 [  4   1   1  14   5   1   8   0 798   3]
 [  9   0   7  21  50   0   1  11  55 684]]
```



Misclassification might have occurred due to less regularization of data. By adding a penalty to the cost function, overfitting can be discouraged. Also to improve the accuracy, the technique of cross-validation can be used that involves splitting your data into multiple sets, and then training your model on one set of data and testing it on another set of data. This helps to ensure that your model does not overfit to the data that it was trained on.

3. Naive Bayes

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It relies on the common principle that every pair of features being classified is independent of each other.

The formula for Bayes' theorem is given as:

$$\mathrm{P(A|B)}= \frac{P(B|A)P(A)}{P(B)}$$

Where,

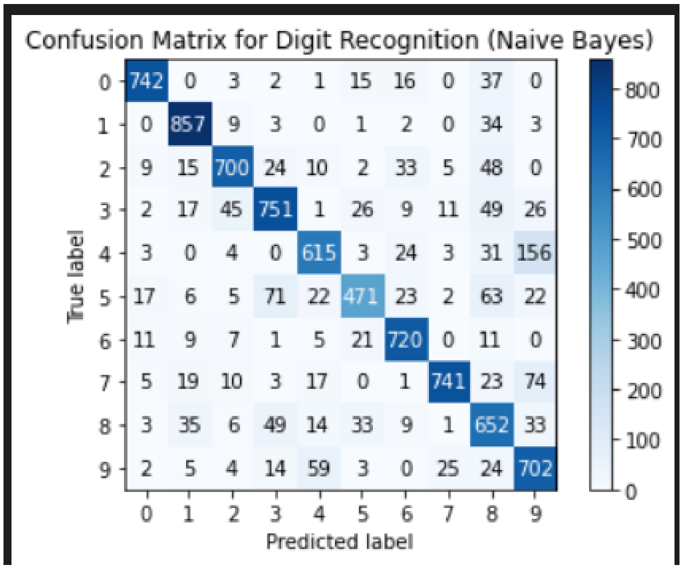
P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

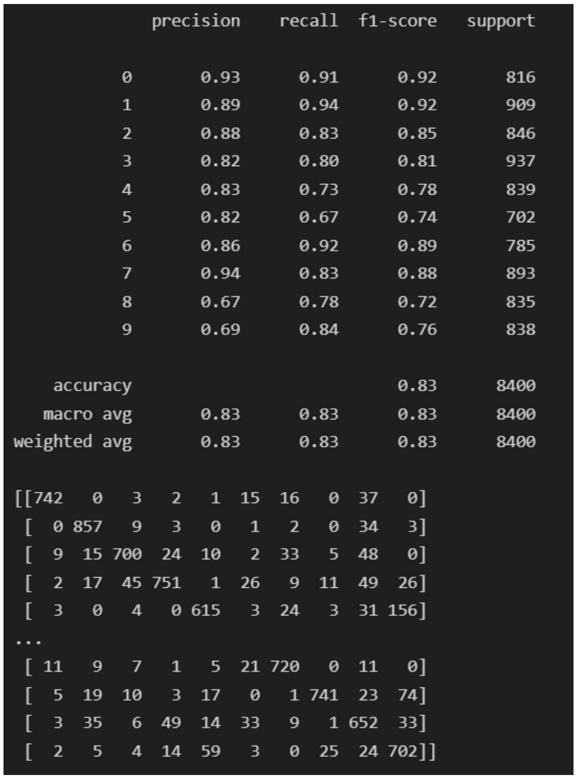
P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

a. Accuracy: 83%
b. Limitations: The algorithm assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

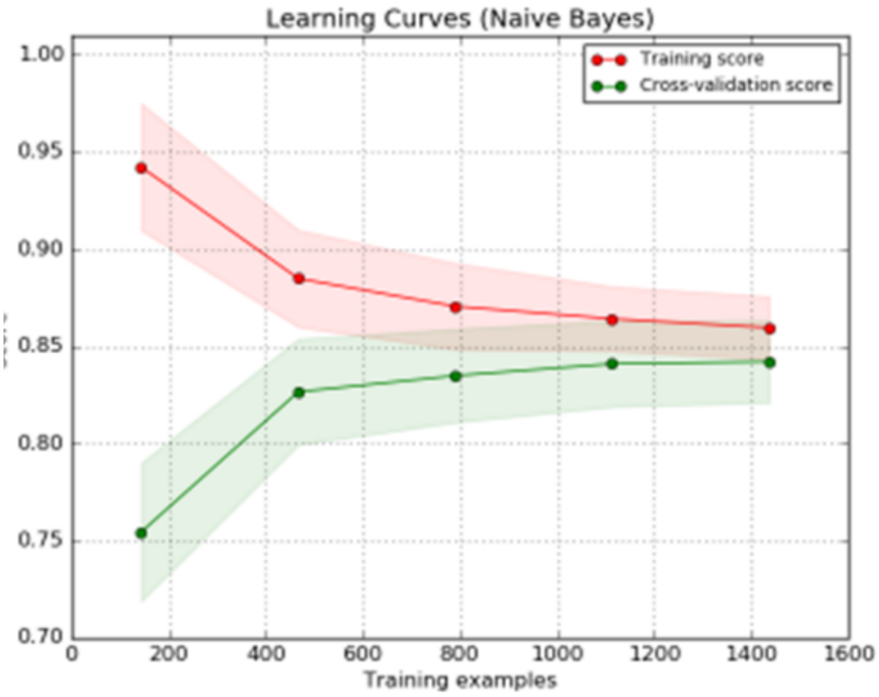
c. Advantages:
   i. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions
   ii. It is suitable for both binary as well as multi-level classification.
   iii. It is highly scalable with the number of predictors and data points.

Accuracy can be improved by reducing the complexity of the model by removing noisy features; For unregularized models, you can use feature selection or feature extraction techniques to decrease the number of features.





```
              precision    recall  f1-score   support

           0       0.93      0.91      0.92       816
           1       0.89      0.94      0.92       909
           2       0.88      0.83      0.85       846
           3       0.82      0.80      0.81       937
           4       0.83      0.73      0.78       839
           5       0.82      0.67      0.74       702
           6       0.86      0.92      0.89       785
           7       0.94      0.83      0.88       893
           8       0.67      0.78      0.72       835
           9       0.69      0.84      0.76       838

    accuracy                           0.83      8400
   macro avg       0.83      0.83      0.83      8400
weighted avg       0.83      0.83      0.83      8400

[[742   0   3   2   1  15  16   0  37   0]
 [  0 857   9   3   0   1   2   0  34   3]
 [  9  15 700  24  10   2  33   5  48   0]
 [  2  17  45 751   1  26   9  11  49  26]
 [  3   0   4   0 615   3  24   3  31 156]
...
 [ 11   9   7   1   5  21 720   0  11   0]
 [  5  19  10   3  17   0   1 741  23  74]
 [  3  35   6  49  14  33   9   1 652  33]
 [  2   5   4  14  59   3   0  25  24 702]]
```



4. **KNN**

   The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

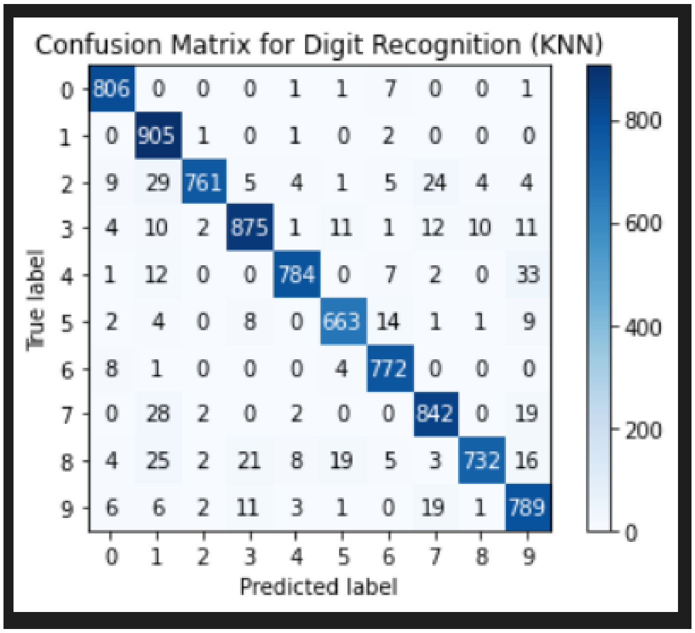
   a. Accuracy: 94%
   b. Limitations:
      i. Does not scale well: It takes up more memory and data storage compared to other classifiers. Costly and time consuming
      ii. Curse of dimensionality: It doesn't perform well with high-dimensional data inputs. Additional features increase the

amount of classification errors, especially when the sample size is smaller.

   iii. Prone to overfitting: While feature selection and dimensionality reduction techniques are leveraged to prevent this from occurring, the value of k can also impact the model’s behavior. Lower values of k can overfit the data, whereas higher values of k tend to “smooth out” the prediction values since it is averaging the values over a greater area, or neighborhood. However, if the value of k is too high, then it can underfit the data.

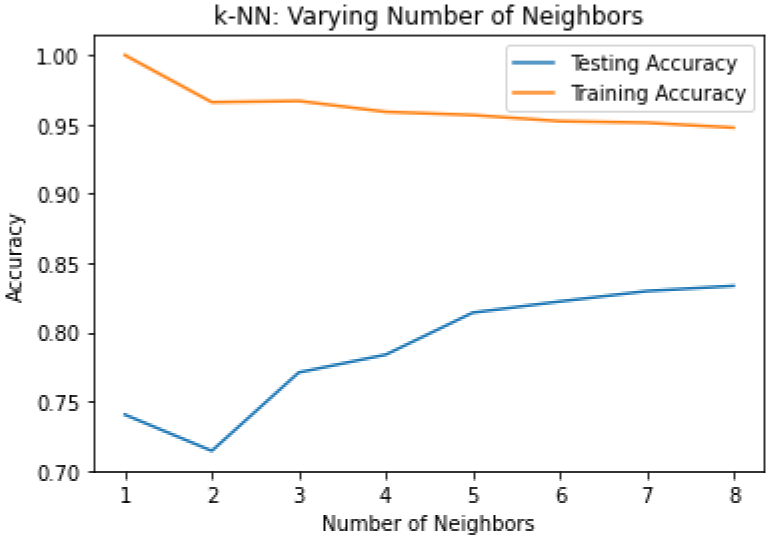
c. Advantages:
   i. Easily implemented
   ii. Easily adaptable: Automatically adjusts for new data to be stored when new set of observations are added
   iii. Few hyperparameters: KNN only requires a k value and a distance metric, which is low when compared to other machine learning algorithms.

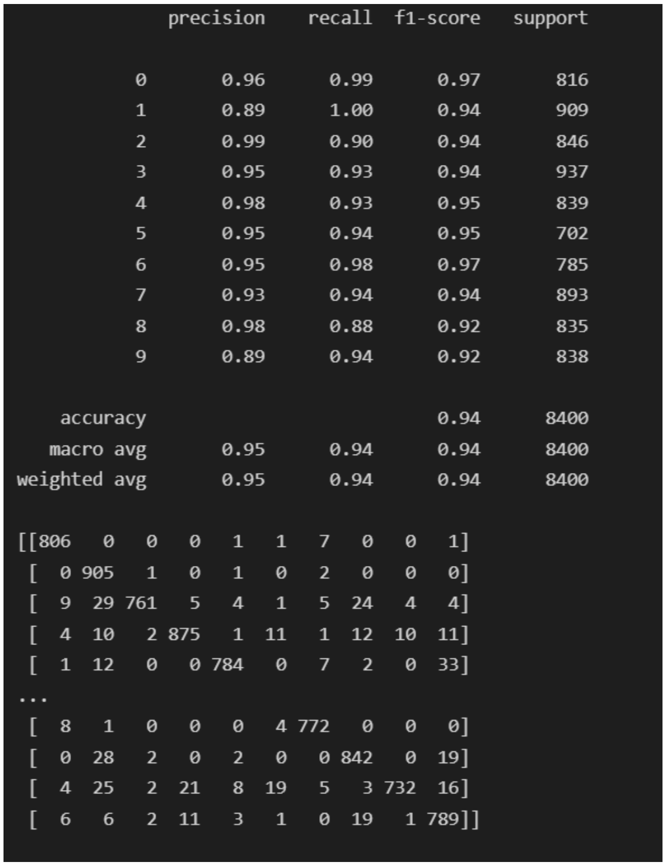
Accuracy can be further improved by using algorithmic tuning that is by parameter tuning. The objective of parameter tuning is to find the optimum value for each parameter to improve the accuracy of the model.



Confusion Matrix for Digit Recognition (KNN)

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	806	0	0	0	1	1	7	0	0	1
1	0	905	1	0	1	0	2	0	0	0
2	9	29	761	5	4	1	5	24	4	4
3	4	10	2	875	1	11	1	12	10	11
4	1	12	0	0	784	0	7	2	0	33
5	2	4	0	8	0	663	14	1	1	9
6	8	1	0	0	0	4	772	0	0	0
7	0	28	2	0	2	0	0	842	0	19
8	4	25	2	21	8	19	5	3	732	16
9	6	6	2	11	3	1	0	19	1	789





```
              precision    recall  f1-score   support

           0       0.96      0.99      0.97       816
           1       0.89      1.00      0.94       909
           2       0.99      0.90      0.94       846
           3       0.95      0.93      0.94       937
           4       0.98      0.93      0.95       839
           5       0.95      0.94      0.95       702
           6       0.95      0.98      0.97       785
           7       0.93      0.94      0.94       893
           8       0.98      0.88      0.92       835
           9       0.89      0.94      0.92       838

    accuracy                           0.94      8400
   macro avg       0.95      0.94      0.94      8400
weighted avg       0.95      0.94      0.94      8400

[[806   0   0   0   1   1   7   0   0   1]
 [  0 905   1   0   1   0   2   0   0   0]
 [  9  29 761   5   4   1   5  24   4   4]
 [  4  10   2 875   1  11   1  12  10  11]
 [  1  12   0   0 784   0   7   2   0  33]
...
 [  8   1   0   0   0   4 772   0   0   0]
 [  0  28   2   0   2   0   0 842   0  19]
 [  4  25   2  21   8  19   5   3 732  16]
 [  6   6   2  11   3   1   0  19   1 789]]
```

As we vary the value of K-Nearest neighbors, the training and testing dataset accuracy score is calculated and thus printed as the accuracy graph comparing training and testing accuracy score. We see here that for the training dataset with increase in KNN, accuracy drops while it is the other way for Testing set indicating a significant improvement.

The confusion matrices for all 4 algorithms provides us the value if true positives, true negatives, false positives and false negatives. By comparing two confusion matrices, we can determine the true positive value let's say for the number 1 is 905 (KNN), 857 (Naive Bayes), 580 (SVM) and 786 (Logistic) that means KNN gives the most number of true 1's for label 1 and hence more accurate. Similarly we can calculate other parameters for different labels thereby getting a sense of the accuracy score, precision and recall.

Bias Variance tradeoff:

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.

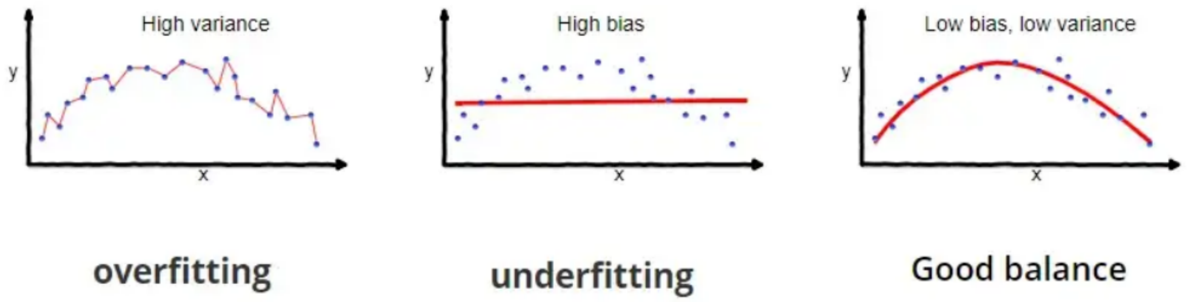
Variance is the variability of model prediction for a given data point or a value which tells us spread of our data.

The algorithms used above all are supervised learning algorithms since the output label data has already been provided to us and we need to classify based on that. Hence most of these algorithms face the limitation of overfitting that happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy datasets. These models have low bias and high variance. Learning curves give us an opportunity to diagnose bias and variance in supervised learning models.
If the training error is high, it means that the training data is not fitted well enough by the estimated model. If the model fails to fit the training data well, it means it has high bias with respect to that set of data.

variance

A narrow gap indicates low variance. Generally, the more narrow the gap, the lower the variance. The opposite is also true: the wider the gap, the greater the variance.

We should always choose hyperparameters so that both bias and variance are as low as possible and more training dataset.
Total Error = Bias + Variance + Irreducible Error



Result:

- Out of the four algorithms (Naive Bayes, Logistic Regression, SVM and KNN) KNN outperformed with an accuracy of about 94%.
- Naive Bayes and KNN took comparatively less time.

Learnings of ML:

1. Some algorithms work better for certain datasets and not for others. There is no one size fits all
2. Overfitting and underfitting is a huge issue that needs to be be tacked well
3. When in doubt - normalize. Most issues can be solved by feature rescaling
4. Optimal and accurate parameter tuning can increase accuracy by bounds. Careful study and understanding of the model can lead us to predict the right hyperparameters.

Resources used:

1. https://www.ibm.com/in-en/topics/knn
2. https://arxiv.org/pdf/2106.12614.pdf
3. https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229
4. https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/#:~:text=SVM%20Disadvantages&text=Long%20training%20time%20for%20large,to%20incorporate%20our%20business%20logic.
5. https://vitalflux.com/learning-curves-explained-python-sklearn-example/
6. https://zahidhasan.github.io/2020/10/13/bias-variance-trade-off-and-learning-curve.html