# Machine Learning Assignment-1

Harsh Wardhan
*Electrical Engineering*
*Indian Institute of Technology, Delhi*
ee3180610@iitd.ac.in

Shreyansh Choudhary
*Maths and Computing*
*Indian Institute of Technology, Delhi*
mt6180794@iitd.ac.in

***Abstract*—This report comprises the results of the various experiments we did on the provided data-sets to obtain best fit models. We have used various techniques to predict the pattern in data, explained the results.**

***Index Terms*—Bayes Classifier, Parametric Estimation, Non-Parametric Estimation, Generalised Linear Models, Logistic Regression, Regularization**

## I. Question1

### A. Problem Statement

A patient is diagnosed for heart disease. Following details of the user are considered to be effective in determining the disease- Age of the patient, Resting BP, serum cholesterol. Task is to obtain a model trained on 700 samples of such patients and predict the chances of heart disease for a completely unknown patient.

### B. Data Pre-processing

We observed that the different measures lies in different ranges. So to avoid problems like underflow/overflow/floating point Errors, we have normalised the data to lie in the same range. Data is shuffled randomly and split into test and training sets.

### C. Metrics Used

- Accuracy, Precision, Recall, F1 Score. Also, different RoC curves are plotted.

### D. Techniques Used

- Gaussian Distribution with unknown mean and unknown variance:
  We plotted the Test Accuracy vs Threshold plot to obtain a good threshold(Fig. 1).
  Best Accuracy is obtained when threshold is 0.45.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 0.45 | 0.88 | 0.81 | 0.9 | 0.857 |
| 1 | 0.867 | 0.79 | 0.95 | 0.86 |

  The RoC Curve for the above model is shown in Fig. 2.
- Gaussian Distribution with unknown mean and same variance:
  Threshold is set to 1.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 1 | 0.8478 | 0.76 | 0.92 | 0.835 |

- Gaussian Mixture Models:
  We Choose k i.e. number of modes of the distribution as

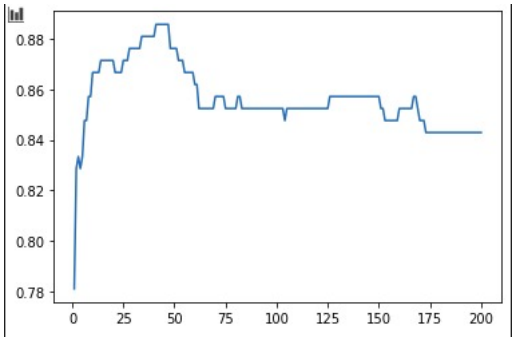


Fig. 1. Accuracy vs Threshold

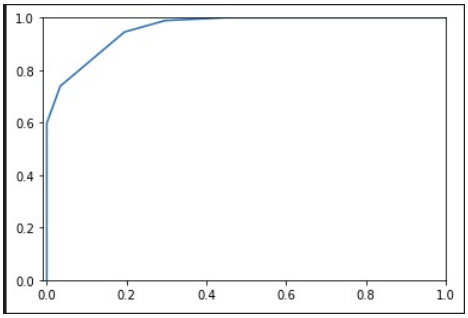


Fig. 2. RoC for Gaussian with Different Co-variance.

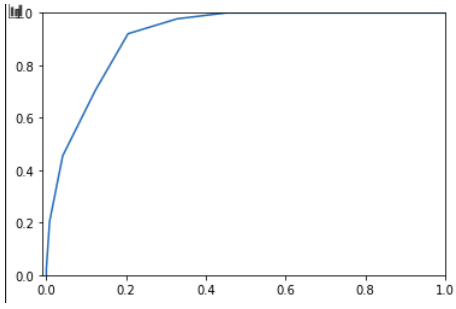


Fig. 3. RoC for Gaussian with same Co-variance.

the model complexity and plotted bias variance curves to find an optimal k(Fig. 3)
We got the minima of the test loss(0-1 loss) at k=1 and plotted the RoC curves for the model.

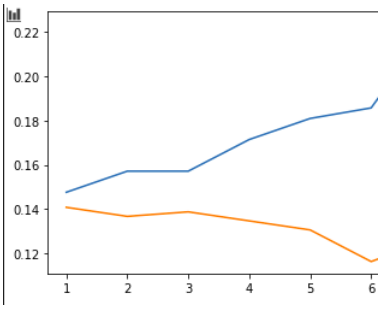


Fig. 4. Bias Variance Curve for GMM. Here the blue plot represents the Test Loss and Orange represents Train Loss.

| Model Performance | | | | |
|---|---|---|---|---|
| k | Accuracy | Precision | Recall | F1 Score |
| 1 | 0.866 | 0.94 | 0.79 | 0.86 |

By comparing the accuracy for the Gaussian Model(MLE) and GMM, k=1, we see that although it is an iterative approach, EM algorithm converged well to the global minima.

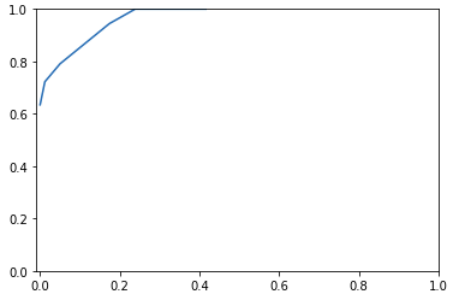


Fig. 5. RoC for GMM, k=1.

- Naive Bayes with IID Gaussian:
  Each Feature is assumed to be IID normally distributed and maximum likelihood estimate is obtained. RoC is plotted for this in Fig. 6

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 1 | 0.871 | 0.8714 | 0.924 | 0.863 |

  Since we observed a better F1-score for Naive Bayes as compared to Gaussian Model(Bayes), we can infer that the IID assumption holds pretty well.
- Non-Parametric Methods
  - K-Nearest Neighbours:
    Euclidean Distance metric was used. To select an optimal value of k, we plot Test Accuracy vs k and find the maxima, there. The plot is shown in Fig. 7. From the plot, it can be observed that k=40 gives highest accuracy.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 1 | 0.8571 | 0.7954 | 0.8536 | 0.8235 |

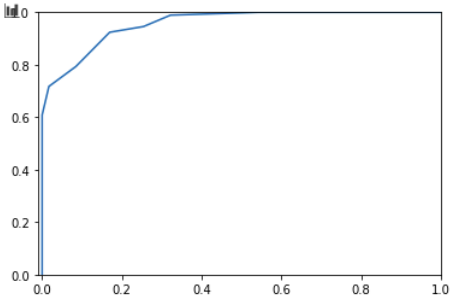


Fig. 6. RoC for Naive Bayes Classifier.

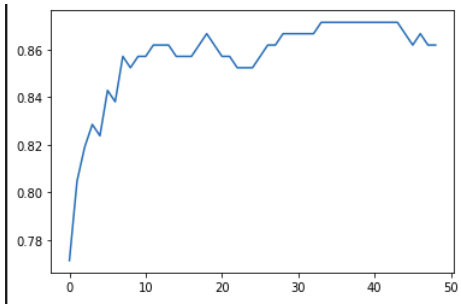


Fig. 7. Accuracy of k-NN vs k.

- Parzen Window Density estimation In this approach the window function was chosen to be hyper-cube of length k. Again k is chosen by plotting Accuracy vs k. (Fig. 9). Euclidean Distance metric was used.

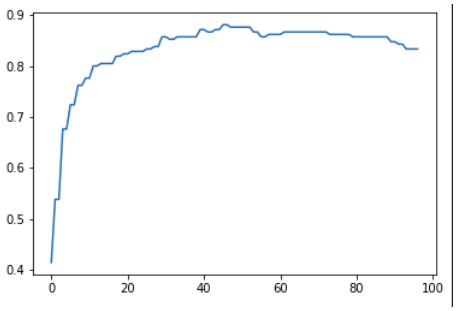


Fig. 9. Accuracy vs window size

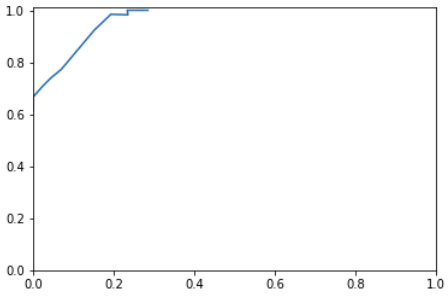


Fig. 8. RoC for k-NN, k=40

The maximum value of Accuracy is observed when window size is 43.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 1 | 0.87 | 0.83 | 0.8656 | 0.85 |

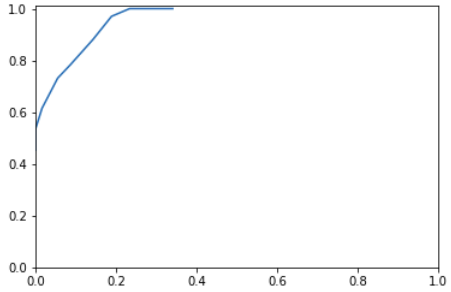


Fig. 10. RoC for Parzen Window, window size=43

- Linear Models
  - When Loss is MAE:
    We choose MAE loss to determine the coefficients. Model complexity is the degree of the polynomial. Bias Variance Curve is plotted against degree of the polynomial. We initialise the W matrix randomly, converge it 10 times and then report minimum test loss. The reason we do this is there are several local minima in the loss function which the gradient descent can converge to. We could have altered the learning rate to help it but it diverges for higher learning rate and converges for sure in some local minima in smaller learning rate.

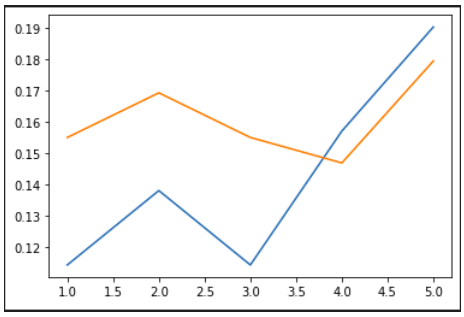


Fig. 11. Bias Variance Curve for Polynomial Regression(MAE loss). Here the blue plot represents the Test Loss and Orange represents Train Loss.

**Observations:** The training loss is not showing ideal behavior, neither the test loss. As explained earlier the train-loss and test-loss won't behave ideally because the gradient descent converges to local minima instead of global one in-spite of initialising it randomly multiple times.
Bias is not decreasing with increasing model complexity.
Selecting degree of the polynomial to be 3 i.e. the minima of the test loss.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 0 | 0.871 | 0.923 | 0.810 | 0.862 |

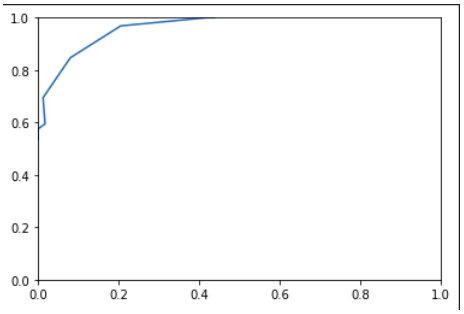


Fig. 12. RoC for Polynomial Regression, Degree=3.

For Regularization, we have chosen the polynomial degree to be 3 and now we use L1, L2, elastic net to prevent over-fitting. We report F1 score for various values of penalty. L1 Regularization:

| F1 Score | |
|---|---|
| $\lambda_1$ | F1 Score |
| 0.25 | 0.8677 |
| 0.5 | 0.739 |
| 0.75 | 0.858 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = 0.25$. L2 Regularization:

| F1 Score | |
|---|---|
| $\lambda_2$ | F1 Score |
| 0.25 | 0.8736 |
| 0.5 | 0.867 |
| 0.75 | 0.869 |

From the above table we can see that the model attains highest F1 score when $\lambda_2 = 0.25$.
Elastic Net Regularization:

| F1 Score | | | |
|---|---|---|---|
| $\lambda_1 \downarrow, \lambda_2 \rightarrow$ | 0.25 | 0.5 | 0.75 |
| 0.25 | 0.863 | 0.853 | 0.869 |
| 0.5 | 0.853 | 0.873 | 0.860 |
| 0.75 | 0.858 | 0.829 | 0.847 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = \lambda_2 = 0.5$.

- When loss is MSE:
  We choose MSE loss to determine the coefficients. Model complexity is the degree of the polynomial. Bias Variance Curve is plotted against degree of the polynomial. We initialise the W matrix randomly, converge it multiple times and then report minimum test loss. The reason we do this is there maybe several local minima in the loss function which the gradient descent can converge to.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 0 | 0.880 | 0.8913 | 0.845 | 0.867 |

For Regularization, we have chosen the polynomial degree to be 3 and now we use L1, L2, elastic net to prevent overfitting. We report F1 score for various values of penalty.

L1 Regularization:

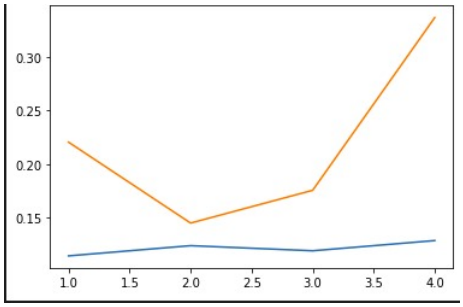


Fig. 13. Bias Variance Curve for Polynomial Regression. Here the orange plot represents the Test Loss and blue represents Train Loss.

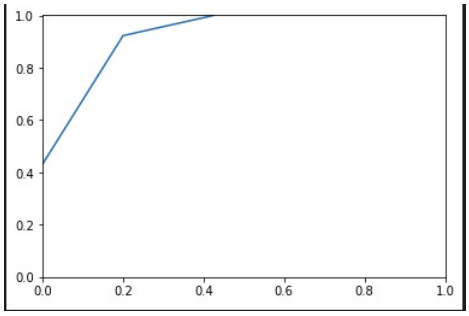


Fig. 14. RoC for Polynomial Regression, Degree=3.

| F1 Score | |
|---|---|
| $\lambda_1$ | F1 Score |
| 0.25 | 0.880 |
| 0.5 | 0.873 |
| 0.75 | 0.771 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = 0.25$

L2 Regularization:

| F1 Score | |
|---|---|
| $\lambda_2$ | F1 Score |
| 0.25 | 0.869 |
| 0.5 | 0.834 |
| 0.75 | 0.779 |

From the above table we can see that the model attains highest F1 score when $\lambda_2 = 0.25$

Elastic Net Regularization:

| F1 Score | | | |
|---|---|---|---|
| $\lambda_1 \downarrow, \lambda_2 \rightarrow$ | 0.25 | 0.5 | 0.75 |
| 0.25 | 0.830 | 0.805 | 0.847 |
| 0.5 | 0.8488 | 0.870 | 0.866 |
| 0.75 | 0.840 | 0.834 | 0.825 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = \lambda_2 = 0.5$.

- Logistic Regression

  We choose cross-Entropy loss to determine the coefficients. Model complexity is the degree of the polynomial. Bias Variance Curve is plotted against degree of the polynomial. We initialise the W matrix randomly, converge it 10 times and then report minimum test loss. The reason we do this is there are several local minima in the loss function which the gradient descent can converge to.

Also for other Loss functions like MSE and MAE, the Risk function is not convex, hence gradient descent might fail miserably.

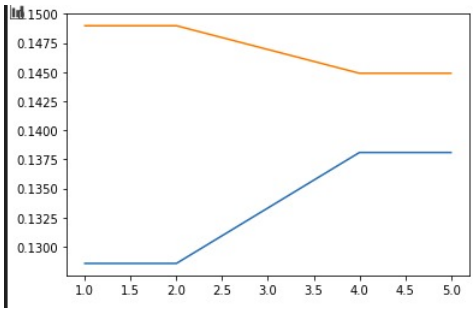


Fig. 15. Bias Variance Curve for Logistic Regression. Here the blue plot represents the Test Loss and Orange represents Train Loss.

| Model Performance | | | | |
|---|---|---|---|---|
| Threshold | Accuracy | Precision | Recall | F1 Score |
| 0.5 | 0.8667 | 0.7753 | 0.8961 | 0.8317 |

For Regularization, we have chosen the polynomial degree to be 3 and now we use L1, L2, elastic net to prevent overfitting. We report F1 score for various values of penalty.

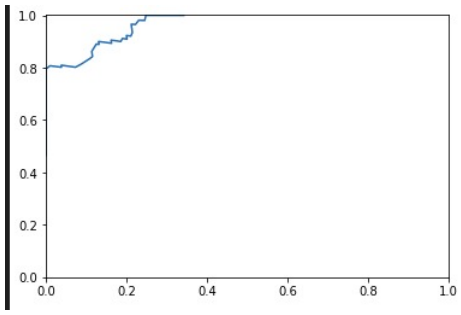


Fig. 16. RoC for Logistic Regression,

L1 Regularization:

| F1 Score | |
|---|---|
| $\lambda_1$ | F1 Score |
| 0.25 | 0.8383 |
| 0.5 | 0.83132 |
| 0.75 | 0.83139 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = 0.75$

L2 Regularization:

| F1 Score | |
|---|---|
| $\lambda_2$ | F1 Score |
| 0.25 | 0.8242 |
| 0.5 | 0.817 |
| 0.75 | 0.8148 |

From the above table we can see that the model attains highest F1 score when $\lambda_2 = 0.25$

Elastic Net Regularization:

| F1 Score | | | |
|---|---|---|---|
| $\lambda_1 \downarrow, \lambda_2 \rightarrow$ | 0.25 | 0.5 | 0.75 |
| 0.25 | 0.82427 | 0.8170 | 0.8148 |
| 0.5 | 0.8242 | 0.8098 | 0.8148 |
| 0.75 | 0.82424 | 0.8172 | 0.8148 |

From the above table we can see that the model attains highest F1 score when $\lambda_1 = \lambda_2 = 0.25$.

### E. Conclusion

- Since the Data turned out to be unimodal by observing bias variance curve of GMM of accuracy v/s number of gaussians in modal we see know that Bayesian Gaussian is clearly better than GMM due to its analytical solution.
- We observe that naive bayes Gaussian outperform Bayes thus we can conclude that features are mutually independent as result confirms our assumption.
- We observe that the data has 2nd Order decision boundary by looking at Bais-Variance curves of Linear models.

## II. QUESTION 2

### A. Problem Statement

Temperature of a city is to be predicted given several parameters like Dew point, Humidity, wind speed, Air pressure, Rain status, smoke status. Find the best model that can fit the provided Data i.e. with minimum error.

### B. Data Pre-processing

We observed that the different measures lies in different ranges. So to avoid problems like floating point Errors, we have normalised the data to lie in the same range. Training Data and Test Data are obtained by random shuffling and 70-30 split.

### C. Metrics Used

- Training Loss and Test Loss, $R^2$ i.e. Coefficient of Determination. Bias Variance Curves are plotted.

### D. Techniques Used

We have implemented Generalised Linear Models with different loss functions and various regularizers. We choose model complexity to be the degree of the polynomial of the 6 features of the data.

- Using Normal Equations(MSE loss):
  Normal Equation method has zero truncation error as it is a non iterative method. Varied degree of polynomial to obtain results.

| Model Performance | | | | |
|---|---|---|---|---|
| Degree | Train loss | Test loss | Train R2 | Test R2 |
| 2 | 0.7108 | 0.152 | 0.991 | 0.997 |

- Using Gradient Descent with MAE loss:
  Ignoring the discontinuity of derivative of modulus function at 0, we applied gradient descent with termination condition being either the values in successive iterations vary by number less than 0.001 or iterations exceed 10000. Learning Rate was chosen to be 0.003.
  We initialise the W matrix randomly, converge it 20 times and then report minimum test-loss. The reason we do this is there are several local minima in the loss function which the gradient descent can converge to. We could have altered the learning rate to help it but it diverges for higher learning rate and converges for sure in some local minima in smaller learning rate.
  We have given 2 bias variance curves One which might capture local minima instead of global Minima(Fig. 17) and another which has the best possible minima among 20 iterations for every degree (Fig. 18)

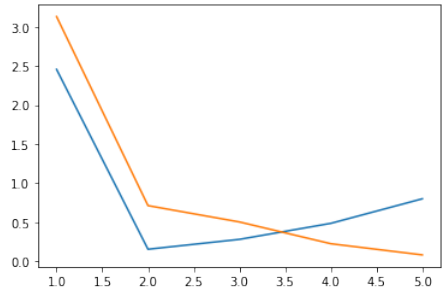


Fig. 17. Bias Variance Curve for Polynomial Regression. Here the blue plot represents the Test Loss and Orange represents Train Loss.

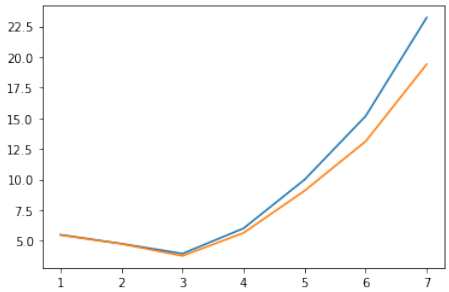


Fig. 18. Bias Variance Curve for Polynomial Regression. Here the orange plot represents the Test Loss and blue represents Train Loss.

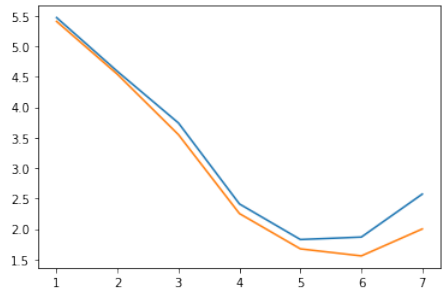


Fig. 19. Bias Variance Curve for Polynomial Regression. Here the orange plot represents the Test Loss and blue represents Train Loss.

| Model Performance | | | | |
|---|---|---|---|---|
| Degree | Train loss | Test loss | Train R2 | Test R2 |
| 6 | 1.86 | 1.55 | 0.843 | 0.934 |

As gradient descent does not converge to global minima, we have not used the regularization. As clearly seen from the graph,the bias does not decrease at all, hence no purpose of applying regularization here. The risk function

is not a convex function, it has several local minima which might attract the descent algorithm.

- Using Gradient Descent with MSE loss:
  Used Gradient descent with termination condition being either the values in successive iterations vary by number less than 0.001 or iterations exceed 100000. Learning Rate was chosen to be 0.007, higher learning rates caused the algorithm to diverge for higher degrees of the polynomial and for lower learning rate the descent ran into local minima.
  We have plotted bias variance graph for polynomial we have obtained minima for test loss for cubic polynomial.

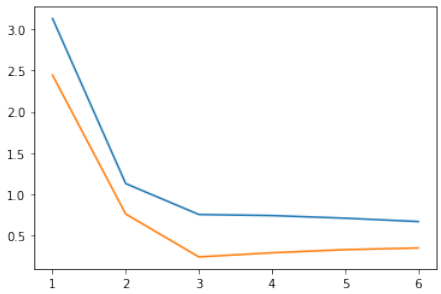


Fig. 20. Bias Variance Curve for Polynomial Regression. Here the orange plot represents the Test Loss and blue represents Train Loss.

| Model Performance | | | | |
|---|---|---|---|---|
| Degree | Train loss | Test loss | Train R2 | Test R2 |
| 3 | 0.7576 | 0.244 | 0.99 | 0.996 |

We have mentioned the results of L1 regularization and L2 regularization in the Elastic Net Regularization table itself. When $\lambda_1$ is 0, the values are for L2 regularization and When $\lambda_2$ is 0, the values are for L1 regularization. Elastic Net Regularization:

| R2 Score | | | | | |
|---|---|---|---|---|---|
| $\lambda_1 \downarrow, \lambda_2 \rightarrow$ | 0 | 0.5 | 1 | 1.5 | 2 |
| 0 | 0.9965 | 0.9961 | 0.9956 | 0.9951 | 0.9947 |
| 0.5 | 0.9966 | 0.9962 | 0.9957 | 0.9953 | 0.9948 |
| 1 | 0.99675 | 0.9962 | 0.9958 | 0.9954 | 0.9949 |
| 1.5 | 0.9968 | 0.9963 | 0.9959 | 0.9955 | 0.9951 |
| 2 | 0.9968 | 0.9964 | 0.9960 | 0.9956 | 0.9952 |

For L1 regularization, we got the maximum $R^2$ score for $\lambda_1 = 1.5$.
For L2 regularization, we got the maximum $R^2$ score for $\lambda_1 = 0.5$
For Elastic-Net regularization, we got the maximum $R^2$ score for $\lambda_1 = 2$ and $\lambda_2 = 0.5$.

## III. Question3

### A. Problem Statement

Given Data comprising several Scans of different body parts. Our task is, given a Scan, predict that with which of the given body parts, it is closely related.

### B. Data Pre-processing

Images of 64x64 pixels are quite big, we have resized the images to 32x32 and then further applied PCA(principal component Analysis) to reduce the number of features.
We observed that the different measures lies in different ranges. So to avoid problems like underflow/overflow/floating point Errors, we have normalised the data to lie in the same range.
We also tried using SIFT operators from opencv-python to extract features through the descriptors, but it couldn't perform well as the kernel it takes is of size 12x12 but our images were of size 64x64. Hence, it couldn't detect any features.
We have used One vs All approach to handle Multi-class Classification.
Classes are serialised alphabetically i.e. AbdomenCT is class 1 and HeadCT is class 6.

### C. Metrics Used

- 5-fold cross validation confusion matrix, per class Precision, per class Accuracy, per class Recall, per class F1-Score, Macro F1 score.

### D. Techniques Used

- Gaussian Distribution with unknown mean and different variance: PCA reduced the features from 1024 to 20.

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| $\downarrow$ Predicted, True $\rightarrow$ | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1594.8 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1424 | 0 | 0 | 0.2 | 0 |
| 3 | 0 | 0 | 1603.8 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1599.8 | 18.6 | 0.4 |
| 5 | 4.4 | 0 | 1 | 8.6 | 1586.2 | 4.4 |
| 6 | 0 | 0 | 0 | 0 | 0.4 | 1586 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| $\downarrow$ Predicted, True $\rightarrow$ | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1984, | 7 | 00 | 2 | 1 | 4 |
| 2 | 1 | 1974 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1792 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 2051 | 0 | 0 |
| 5 | 21 | 0 | 0 | 0 | 1996 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1949 |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.996 | 0.988 | 0.9887 | 0.9890 |
| 2 | 0.999 | 0.9994 | 0.997 | 0.9964 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 0.9998 | 1 | 0.9995 | 0.9990 |
| 5 | 0.9973 | 0.9895 | 0.9922 | 0.9950 |
| 6 | 0.9996 | 1 | 0.9987 | 0.9979 |
| Macro F1 Score: 0.9962 | | | | |

We observe that we got very high Macro-F1 Score, indicating that the data is highly clustered around the sample mean of the respective classes.

- Naive Bayes with each Feature IID with Gaussian Distribution with unknown mean and different variance:
  PCA reduced features from 1024 to 20.

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| $\downarrow$ Predicted, True $\rightarrow$ | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1577.2 | 0 | 14.4 | 0 | 0 | 0 |
| 2 | 0 | 1419.2 | 0 | 0 | 0.2 | 96.2 |
| 3 | 0 | 0 | 1590.4 | 0 | 0 | 0 |
| 4 | 9.8 | 0 | 0 | 1589.4 | 75.4 | 0 |
| 5 | 12.2 | 1.4 | 0 | 15.4 | 1498 | 12 |
| 6 | 0 | 0 3.4 | 0 | 3.6 | 31.8 | 1482.6 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1862 | 13 | 3 | 0 | 24 | 15 |
| 2 | 40 | 1836 | 5 | 0 | 4 | 0 |
| 3 | 1 | 132 | 1784 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 2034 | 0 | 0 |
| 5 | 103 | 0 | 0 | 0 | 1978 | 18 |
| 6 | 0 | 0 | 0 | 19 | 0 | 1920 |

| per-Class Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.983 | 0.9713 | 0.9492 | 0.9282 |
| 2 | 0.9835 | 0.9740 | 0.9498 | 0.9268 |
| 3 | 0.9880 | 0.9306 | 0.9619 | 0.9955 |
| 4 | 0.99838 | 1 | 0.9953 | 0.9907 |
| 5 | 0.9873 | 0.9423 | 0.9637 | 0.9860 |
| 6 | 0.9955 | 0.9902 | 0.9866 | 0.9831 |
| Macro F1 Score: 0.9678 | | | | |

It performs worse than the Gaussian Bayesian Model(MLE) because the pixels in images are not independent of each other. Hence, assumption of IID features is wrong.

- Gaussian Mixture Models:
  We have down-sampled the Data to 10000 samples for GMM due to complexity issues of the EM algorithm.
  PCA reduced features from 1024 to 20.
  We have provided data for k=1 because as k increases the model began to over-fit. We provide the test and train F1-Score to show our point.

| k | Train F1-score | Test F1-Score |
|---|---|---|
| 1 | 0.999 | 0.989 |
| 2 | 0.999 | 0.963 |

Hence, the provided Confusion Matrix are for k=1.

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 252.6 | 1.6 | 0 | 0 | 4 | 1 |
| 2 | 3.8 | 281 | 0 | 0 | 0.2 | 0 |
| 3 | 0 | 0 | 248 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 268.8 | 0 | 0 |
| 5 | 7.8 | 0 | 0 | 0 | 261.2 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 270 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 319 | 1 | 0 | 0 | 7 | 0 |
| 2 | 0 | 368 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 286 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 329 | 0 | 0 |
| 5 | 4 | 0 | 0 | 0 | 326 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 360 |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| Class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.994 | 0.9755 | 0.9815 | 0.9876 |
| 2 | 0.9995 | 1 | 0.9986 | 0.9972 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |
| 5 | 0.9945 | 0.9878 | 0.9834 | 0.9789 |
| 6 | 1 | 1 | 1 | 1 |
| Macro F1 Score: 0.9939 | | | | |

As we can see, the model doesn't perform as good as Gaussian Model, this shows the truncation error in the iterative EM algorithm.

- k-Nearest Neighbors:
  We have down-sampled the Data to 20000 samples due to complexity issues and limited computational power.
  with PCA, we reduced the dimension of data points from 1024 to 5 due to limited computational power.

  We have plotted F1-Score vs k graph to select optimal k.

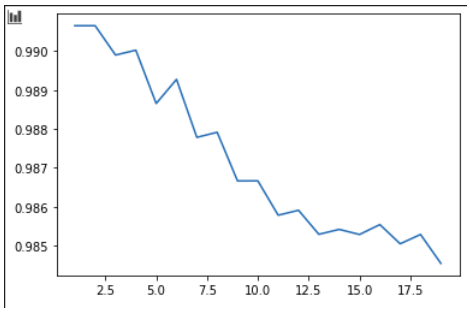


Fig. 21. F1-Score vs K

As we can see, the optimal value of k is 25. We have provided the results for k=25.

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 526.8 | 2 | 0 | 0 | 6 | 0 |
| 2 | 7.6 | 541 | 0 | 0 | 1 | 0 |
| 3 | 0.6 | 0 | 473.4 | 0 | 0. | 0 |
| 4 | 0 | 0 | 0 | 545.8 | 0.2 | 0 |
| 5 | 3.4 | 0 | 0 | 0 | 557.2 | 0. |
| 6 | 0.2 | 0 | 0 | 0 | 0 | 534.8 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 660 | 4 | 0 | 0 | 6 | 0 |
| 2 | 11 | 669 | 0 | 0 | 4 | 0 |
| 3 | 1 | 0 | 598 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 683 | 1 | 0 |
| 5 | 10 | 0 | 0 | 0 | 670 | 0 |
| 6 | 2 | 0 | 0 | 0 | 0 | 681 |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| Class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.9916 | 0.985 | 0.9752 | 0.9656 |
| 2 | 0.9953 | 0.9788 | 0.9863 | 0.9940 |
| 3 | 0.9997 | 0.9983 | 0.9991 | 0.999 |
| 4 | 0.9997 | 0.9985 | 0.9992 | 0.999 |
| 5 | 0.9950 | 0.9860 | 0.9853 | 0.9845 |
| 6 | 0.9995 | 0.9970 | 0.9985 | 0.9976 |
| Macro F1 Score: 0.9906 | | | | |

- Parzen Window:
  We have down-sampled the Data to 20000 samples due to complexity issues and limited computational power.
  We have used hyper-cubes as window function

  We have plotted F1-Score vs window size graph to select optimal k.
  As we can see, the optimal value of window size is 40.
  We have provided the results for window size =40.

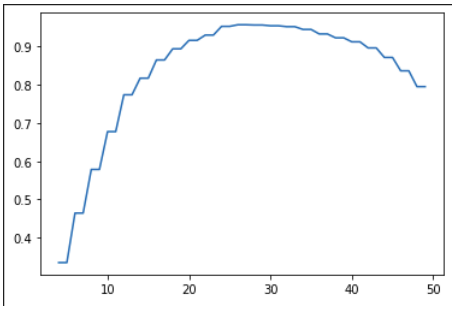


Fig. 22. F1-Score vs Window Size

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 511.4 | 0.4 | 0. | 0 | 0 | 3.8 |
| 2 | 5.6 | 542.2 | 0. | 0. | 0.4 | 0. |
| 3 | 0. | 0. | 473.4 | 0. | 0. | 0. |
| 4 | 0. | 0. | 0. | 545.8 | 0. | 15.6 |
| 5 | 1.4 | 0. | 0. | 0. | 543.4 | 0. |
| 6 | 20.2 | 0.4 | 0. | 0. | 16.8 | 519.2 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 651 | 0 | 0 | 0 | 3 | 0 |
| 2 | 9 | 682 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 567 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 702 | 0 | 27 |
| 5 | 2 | 0 | 0 | 0 | 642 | 0 |
| 6 | 18 | 0 | 0 | 0 | 17 | 680 |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| Class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.9920 | 0.9954 | 0.9760 | 0.9573 |
| 2 | 0.9977 | 0.9869 | 0.9934 | 0.9901 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 0.9932 | 0.9629 | 0.9811 | 0.9719 |
| 5 | 0.994 | 0.9968 | 0.9831 | 0.9697 |
| 6 | 0.98451 | 0.9510 | 0.9563 | 0.9618 |
| Macro F1 Score: 0.9816 | | | | |

Observations : We see that kNN performs better than parzen window which was expected as window function was not appropriate for data is clustered around mean so hyper cube allows much more chances of error.

- Logistic Regression:

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1465.4 | 23.6 | 0 | 0 | 6.4 | 0 |
| 2 | 75 | 1499 | 1.8 | 0 | 4.4 | 7.4 |
| 3 | 2.8 | 67.61 | 1437.8 | 0 | 0 | 0 |
| 4 | 6.8 | 0 | 0 | 1608.6 | 6.60 | 292.4 |
| 5 | 23.4 | 2.20 | 0 | 0 | 1572.2 | 8.8 |
| 6 | 14.2 | 13.4 | 0 | 0.2 | 4.4 | 1288.2 |

| Confusion Matrix on Test Data | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1821 | 41. | 0 | 0 | 3 | 0 |
| 2 | 78 | 1915 | 1 | 0 | 7 | 2 |
| 3 | 6 | 67. | 1816 | 0 | 1 | 0 |
| 4 | 10 | 0 | 0 | 2012 | 13 | 311 |
| 5 | 36 | 3 | 0 | 0 | 1985 | 10 |
| 6 | 9 | 8 | 0 | 1 | 5 | 1630 |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.98441 | 0.97641 | 0.9521 | 0.9290 |
| 2 | 0.9824 | 0.9560 | 0.9487 | 0.9415 |
| 3 | 0.9936 | 0.9608 | 0.9797 | 0.9994 |
| 4 | 0.9716 | 0.8576 | 0.9231 | 0.9995 |
| 5 | 0.9934 | 0.9759 | 0.9807 | 0.9856 |
| 6 | 0.9934 | 0.9759 | 9807 | 0.9856 |
| Macro F1 Score:0.9481 | | | | |

We have used elastic-net regularization and set $\lambda_1 = 0$ for L2 regularization and $\lambda_2 = 0$ for L1 regularization. We present our Selection for $\lambda_1$ and $\lambda_2$ for elastic net regularization in the table below.

| Selection of Regularization parameters | | | | | |
|---|---|---|---|---|---|
| Weights | 0 | 0.5 | 1 | 1.5 | 2 |
| 0 | 0.7361 | 0.8392 | 0.8779 | 0.729 | 0.9074 |
| 0.5 | 0.7781 | 0.855 | 0.8561 | 0.7223 | 0.9385 |
| 1 | 0.7262 | 0.7307 | 0.9285 | 0.7310 | 0.8289 |
| 1.5 | 0.8711 | 0.8640 | 0.8080 | 0.7334 | 0.7705 |
| 2 | 0.9445 | 0.7372 | 0.9417 | 0.8884 | 0.8450 |

Maximum F1-Score was obtained for $\lambda_1 = 2$ and $\lambda_2 = 0$
For L1 regularization, we got the maximum macro F1 score for $\lambda_1 = 2$.
For L2 regularization, we got the maximum macro F1 score for $\lambda_2 = 2$
For Elastic-Net regularization, we got the maximum macro F1 score for $\lambda_1 = 1$ and $\lambda_2 = 1$.

| 5-fold cross validation Confusion Matrix | | | | | | |
|---|---|---|---|---|---|---|
| ↓ Predicted, True → | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1473.4 | 39.6 | 0 | 0 | 6.2 | 0 |
| 2 | 69.2 | 1478.2 | 0 | 0 | 4 | 7.2 |
| 3 | 2.6 | 71. | 1439.6 | 0 | 0.2 | 0 |
| 4 | 5 | 0 | 0 | 1608.8 | 7 | 319.2 |
| 5 | 24 | 5 | 0 | 0 | 1573.2 | 11.4 |
| 6 | 13.4 | 12. | 0. | 0 | 3.4 | 1259. |

| Class Wise Accuracy Precision, Score , Recall | | | | |
|---|---|---|---|---|
| class | Accuracy | Precision | F1 Score | Recall |
| 1 | 0.9644 | 0.9282 | 0.8885 | 0.8520 |
| 2 | 0.9565 | 0.8971 | 0.8686 | 0.8418 |
| 4 | 0.9903 | 0.9502 | 0.9723 | 0.9955 |
| 5 | 0.9921 | 0.9691 | 0.9769 | 0.9848 |
| 6 | 0.9920 | 0.9892 | 0.9762 | 0.9636 |
| Macro F1 Score:0.9380 | | | | |

### *E. Conclusion*

- We observe that Data is unimodal Gaussian by observing Macro F1 test score by observing increment in number of gaussians in GMM.
- The Data is highly clustered as we observe very high macro F1 score in gaussian models.
- Naive Bayes performs much worse than normal Bayes which passes our sanity check as our features are image pixels and they are not independent to each other which is depicted by the results.


## ACKNOWLEDGMENT

Special Thanks to all my mates on Piazza.

## References

[1] https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c. Referred for usage of losses in Logistic regression