Assignment 4

Objective of this assignment is to design multivariate classifiers from first principles. You may choose to either extend the univariate methods from assignment 1, for the multivariate scenarios or design new approaches. However, do not use any off the shelf classification algorithms.

The task of your classifier is gender identification, based on measured parameters. We have a toy data set of 1000 male and 1000 female (labelled) samples. We attempt height based univariate classifier and realize that due to overlap in heights, there are limitation on improving accuracy meaningfully without exploring other features. So, we decide to identify and add new features to our learning algorithms to reduce the prediction error further.

Consider following scenarios of increasing complexity.

1. Uncorrelated input features ( 5 marks)
   1. We have two input features say, Height( in cm ) and Heamoglobin levels measured for all 2000 samples. Let’s assume that both these features are normally distributed within each gender. These features are pretty much uncorrelated within each gender.
   2. Can you design approaches to train a classification algorithms to predict gender?

Since height and haemoglobin levels are normally distributed and uncorrelated, we can take a straightforward probabilistic approach to classify gender.

**Steps:**

* Standardize the height and haemoglobin levels so that both features contribute equally. We can do z-score normalization.
* Compute the mean and standard deviation for each gender separately.
* Assume that males and females have distinct distributions for height and haemoglobin.
* Using Bayes’ theorem, we can compute the probability of a given sample being male or female based on both features.
* If the probability of a sample belonging to the male class is greater than the female class, classify it as male; otherwise, classify it as female.
* Calculate the probability density functions (PDF) for height and haemoglobin separately for each gender.
* Multiply these PDFs to obtain a joint likelihood for each gender and assign the sample accordingly.
* Since the features are uncorrelated, this method should perform well.

1. Input features with non-zero correlation (3 marks)
   1. In this scenario, we have two input features say, Height(in cm) and weight(in kg) measured for all 2000 samples. Both are normally distributed within each gender. The correlation between these features is 0.6 within each gender.
   2. Which of the algorithms you designed for uncorrelated features would work as is? If they don’t, what changes can you make to your algorithms to accommodate correlations.

As height and weight are correlated, we can modify the approach as -

* Instead of treating height and weight independently, we consider their joint distribution using a multivariate normal distribution.
* Compute the mean vector and covariance matrix for each gender.
* Use the Multivariate Gaussian Distribution Formula to estimate the probability of a sample belonging to each gender.
* Assign the class based on which probability is higher.
* The correlation affects how the features interact, and modeling them jointly helps capture their relationship.

2nd method -

1. Compute the Mean Vector and Covariance Matrix for Each Gender:
   * The covariance matrix captures the relationship between height and weight.
   * If height and weight are correlated, the covariance matrix will have non-zero off-diagonal elements.
2. Make Features Independent:
   * Transform the feature space to remove correlation using Principal Component Analysis (PCA).
   * This converts correlated features into uncorrelated principal components.
3. Use Mahalanobis Distance for Classification:
   * Instead of Euclidean distance, use Mahalanobis distance, which accounts for correlations between features.
   * This measures how far a data point is from the mean, adjusting for correlations.
4. Assign the Class Based on the Minimum Mahalanobis Distance:
   * Compute DM(x) for both male and female distributions.
   * Assign the sample to the class with the smaller Mahalanobis distance.

* This approach removes dependency between height and weight, making them act as independent features.
* Mahalanobis distance accounts for correlation, unlike Euclidean distance, which treats features as if they are uncorrelated.

1. How far can we go? (2 marks)
2. We observe that accuracy improves with addition of one new feature in both of the above scenario. Can we reach a conclusion that accuracy can be improved further by adding multiple such features to the input? How many such features would you add in your quest to improve accuracy? Would addition of new features require any changes to the experimental set up?

Adding more features can improve accuracy up to a certain limit before we face diminishing returns or overfitting.

* More Features = More Information: Adding features like hand size, or muscle mass could improve classification accuracy.
  + Adding too many features can lead to overfitting, especially if we don’t have enough training data.
  + Feature Selection Techniques (like PCA) can help reta
  + in only the most relevant features.
  + More features require a larger dataset to maintain generalization.
  + We may need dimensionality reduction techniques if features become redundant.

My observations –

* Uncorrelated features work well with simple probability-based models.
* Correlated features need joint probability modeling.
* Adding more features helps, but too many can lead to overfitting.