Indian Institute of Technology Madras

ID5055 Foundations of Machine learning

Assignment IV

Due date: 14^{th} November 2023

Instruction

1. Assignment shall be submitted on the due date. Late submissions will not be entertained. If you cannot submit the assignment due to some reasons, please contact the instructor by email.
2. All the assignments must be the student's own work. The students are encouraged to collaborate or consult friends. In the case of collaborative work, please write every student's name on the submitted solution.
3. If you find the solution in the book or article or on the website, please indicate the reference in the solutions.

Problems

1. [Decision Tree] Consider the following dataset of the students of class ID5055. Using this dataset, construct a decision tree model(multi-way split) using information gain criteria to predict if the student will ace the end-semester exam, based on their mid-semester performance and if they have submitted all their assignments. Please ensure you show all the necessary calculations.

Mid Sem Performance	**Submitted Assignments**	**Aced End Sem**
Below Average	False	False
Below Average	False	False
Below Average	True	True
Below Average	False	False
Average	True	False
Average	True	True
Average	False	False
Average	True	True
Average	True	True
Above Average	False	True
Above Average	True	True

2. [Logistic Regression] Recall that learning logistic regression model using Newton-Raphson method for a 2-class classification problem reduces to iteratively updating updating $\bar{\beta}$ until convergence:

$$\bar{\beta}_{new} = \bar{\beta}_{old} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\bar{y} - \bar{P})$$

Here, $\bar{\beta}$, $\mathbf{X}$, $\bar{y}$, $\bar{P}$, $\mathbf{W}$ are as defined in our LR lecture slides.

Show that the above solution can also be obtained from minimizing weighted least square problem:

$$\bar{\beta}_{new} = arg \min_{\bar{\beta}} (\bar{z} - \mathbf{X}\bar{\beta})^T \mathbf{W} (\bar{z} - \mathbf{X}\bar{\beta}),$$

where, $\bar{z} = \mathbf{X}\bar{\beta}_{old} + \mathbf{W}^{-1}(\bar{y} - \bar{P})$

3. [QUADRATIC DISCRIMINANT ANALYSIS] In our class we derived the expression for the class boundary and class prediction for LDA. Similary, derive the expressions for QDA and show that the class boundary is quadratic.
Note: QDA assumes that the class conditional probability, $Pr(\bar{x}|Y = k)$, is gaussian with distinct mean and covariance.

4. [NAIVE BAYES] Apply Naive Bayes on the same dataset as defined in question 1 and estimate the probability of a student excelling in the end-semester exam given her mid-semester performance is "Average" and has submitted all her assignments.

5. [NAIVE BAYES] Consider a Gaussian Naive Bayes classifier for a dataset with single attribute x and two classes 0 and 1. The parameters of the gaussian distributions are:

$$p(x|y = 0) \sim \mathcal{N}(0, 1/4)$$
$$p(x|y = 1) \sim \mathcal{N}(0, 1/2)$$
$$p(y = 1) = 0.5$$

Find the decision boundary for the classifier if the loss matrix is $L = \begin{bmatrix} 0 & \sqrt{2} \\ 1 & 0 \end{bmatrix}$.

6. [SUPPORT VECTOR MACHINES] In the SVM formulation, let us say you wanted to optimize over $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ such that the prediction function is of the form $\hat{\mathbf{y}} = sign(\mathbf{w}^T x + b)$ with the additional bias term.
   (a) Derive the bias b for the hard-margin SVM problem.
   (b) Derive the bias b in the case of soft-margin SVM.

7. [SUPPORT VECTOR MACHINES] The following short questions should be answered with appropriate explanations.
   (a) Consider a point that is correctly classified and distant from the decision boundary. Why would SVM's decision boundary be unaffected by this point, but the one learned by logistic regression be affected?
   (b) Why does the kernel trick allow us to solve SVMs with high dimensional feature spaces, without significantly increasing the running time?

8. [RANDOM FOREST CLASSIFIER] Consider the traffic prediction dataset provided here for predicting the traffic condition as 'normal','low','heavy' and 'high'.
   (a) Split the dataset into train and test set (train_size= 0.8, random_state = 42) and train a random forest classifier using the train set. Plot a confusion matrix using the test set for prediction.
   (b) Use a weighted random forest classifier with weights based on the frequency of the corresponding class. Plot a confusion matrix and report your observation by comparing the results with the previous results.
   (c) Use the trained classifier model to report the important features based on impurity metric.