Supplementary material for ML coursework:

I. Glossary

LR – Logistic Regression
NB – Naïve Bayes
TP – True Positive
FP – False Positive
FN – False Negative
TN – True Negative
_hp – suffix ending with _hp is hyperparameter optimization model (eg: TP_hp)

II. Intermediate results including any negative results

Let me point out the main error we faced and overcome,

1. While building model in LR using mnrfit(X,Y), X should be an array type, Y should be an categorical type or if it's a int/float/double type then it should be mentioned as hierarchical like this B = mnrfit(X,Y,'Model','hierarchical') in that case Y vector should have positive numbers, 0 may affect the model by getting multipled with another array.
2. Using table2array, because many of the function calls for an array, where tables will not be accepted, there we used it.

III. Implementation details including a brief description of the main implementation choices

A. First let's see the implementation of Logistic Regression:

1. Load Data:
   a. Load data using: *data=readtable()*
   b. Load label encoded data also. Because LR works with only numeric values.

2. Data partition:
   a. Using cvpartition – KFold method in order to do a KFold cross validation of model with training dataset.
   b. Split data in to X and Y set

3. Logistic Regression:
   a. Use a for loop to run the cross validation for 10 times in the train dataset.
   cv = cvpartition(4119, 'KFold', 10);

   b. Split datasets into test and train dataset

   c. Use the below function to train the model
   model_lr = mnrfit(X_train, Y_train, 'Model', "hierarchical");

   d. Use the below function for evaluating the model, where pihat is a 2-column data which store the max liklehood probability of the class 'yes' and 'no'
   pihat = mnrval(model_lr, X_test);

   e. Compare both the probability of the pihat and the max has to be chosen
   [prob,yihat] = max(pihat,[],2);

   f. Run a cm = confusionmat(Y_test, yihat); to get the TP, FP FN, TN.

   g. Accuracy, Precision, Recall, F1 Score will be derived from TP, FP FN, TN.

4. Plot ROC curve:
   a. Plot ROC using [fpr,tpr,thresh]= perfcurve(Y_test,pihat(:,1),1);

5. Confusion Matrix Chart
   a. This plots the confusion matrix. cmc = confusionchart(Y_test,yihat);

6. Plot Metrics for CV
   a. Plot Accuracy, Precision, Recall, F1 Score of the cross validation array.

B. Implementation of Naïve Bayes:

1. Load Data:
   a. Load data using: *data=readtable()*
      Naïve Bayes accepts categorical data.

2. Data partition:
   a. Using cvpartition splitting the data into X and Y set

3. Naïve Bayes Regression:
   a. Split datasets into test and train dataset using the below function
      cv = cvpartition(n,'HoldOut', 0.2);

   b. Use the below function to train the model
      model = fitcnb(X_train,Y_train);

   c. Use the below function to predict Y,
      [Y_predict,scores]= predict(model,X_test);

   d. Run the below code to get the TP, FP FN, TN.
      cm = confusionmat(table2array(Y_test), Y_predict)

   e. Accuracy, Precision, Recall, F1 Score will be derived from TP, FP FN, TN.

4. Plot ROC curve:
   a. Plot ROC using the below function
      rocObj = rocmetrics(table2array(Y_test),scores, model.ClassNames);

5. Confusion Matrix Chart
   a. This plots the confusion matrix.
      cmc = confusionchart(table2array(Y_test),Y_predict);

B.1 Implementation of Naïve Bayes - hyperparameter optimization:

The type of naive Bayes classifier: There are several types of naive Bayes classifiers, including Gaussian naive Bayes, Bernoulli naive Bayes, and Multinomial naive Bayes. Each of these classifiers makes different assumptions about the distribution of the data, and may be more or less suitable for different types of data.

The smoothing parameter: The smoothing parameter is used to smooth the probability estimates of the classifier, which can help to prevent overfitting. A common smoothing parameter is called "alpha," which is used in the Multinomial naive Bayes classifier.

6. Naïve Bayes Regression with hyperparameter optimization:
    a. Use the below function to train the model
    model_hp = fitcnb(X_train,Y_train,'OptimizeHyperparameters','auto');

    b. Use the below function to predict Y,
    Y_predict_hp = predict(model_hp,X_test);

    c. Run the below code to get the TP, FP FN, TN.
    cm_hp = confusionmat(table2array(Y_test), Y_predict_hp);

    d. Accuracy, Precision, Recall, F1 Score will be derived from TP, FP FN, TN.

7. Plot ROC curve:
    a. Plot ROC using the below function
    rocObj = rocmetrics(table2array(Y_test),scores_hp, model.ClassNames);
    plot(rocObj, ClassNames=model.ClassNames(1))

8. Confusion Matrix Chart
    a. This plots the confusion matrix.
    cmc = confusionchart(table2array(Y_test),Y_predict_hp);