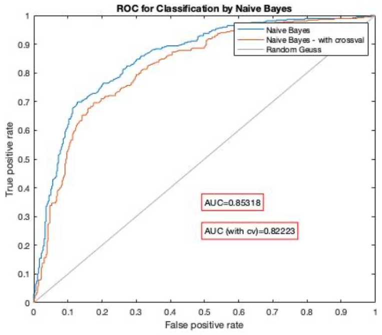
Machine Learning Coursework - Supplementary Material

Babak Hessamian

Intermediate results:

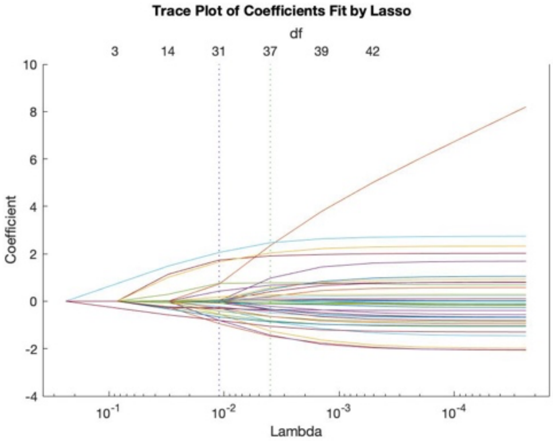
- The parameters that we could try for NB were limited because of mix of data types. One example is the comparison between a cross-validated NB model and a non cross-validated NB model. This did not make much of difference:

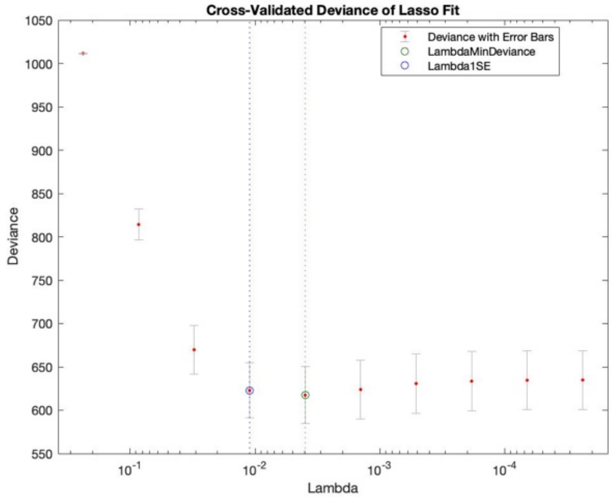


- Using any distribution other than mvmn would not work with NB. However, corss-validating looks at both ‘normal’ and ‘mvmn’ distributions. The mix of categorical and numerical values is really
- Step-wise regression was used to improve the performance of our LR model. This took too long and was used for feature selection.
    - Stepwise code:

```
-   %% Using stepwiseglm to remove varibales to imporve the performance
-   mdl2 = stepwiseglm(XTrain,yTrain,'constant','Distribution','binomial','Upper','linear');
    %https://uk.mathworks.com/help/stats/generalized-linear-model-workflow.html
-   plotDiagnostics(mdl2,'leverage');
-
-   idxOutliers = find(mdl2.Diagnostics.Leverage > 5*mdl2.NumCoefficients/mdl2.NumObservations);
-
-   oldCoeffs = mdl2.Coefficients.Estimate;
-
-   mdl3 = fitglm(XTrain,yTrain,'Distribution','binomial','Link','logit', 'Exclude',idxOutliers);
-   scores = mdl3.Fitted.Probability;
-   [X,Y,T,AUC] = perfcurve(yTrain,scores,'1');
-   AUC
-   plot(X,Y)
-   xlabel('False positive rate')
-   ylabel('True positive rate')
-   title('ROC for Classification by Logistic Regression')
-
-   %mdl3 = fitglm(XTrain,yTrain,'linear','Distribution','binomial', ...
-   %    'PredictorVars','Exclude',idxOutliers);
-   newCoeffs = mdl3.Coefficients.Estimate;
-
-   disp(oldCoeffs)
-   disp(newCoeffs)
-
```

- Performed lasso regularisation on our LR model, but did not drop the features. Some of this is shown in the LR-lasso script:





- The class imbalance posed a real problem. Tried using Synthetic Minority Oversampling Technique (SMOTE) to compare results, but this proved too difficult and was not covered in the module.

Below is some of my attempted code for different steps

Manual regression code:

```
%link = @(mu) log(mu ./ (1-mu));  % https://uk.mathworks.com/help/stats/glmfit.html
%derlink = @(mu) 1 ./ (mu .* (1-mu));
%invlink = @(resp) 1 ./ (1 + exp(-resp));
%F = {link, derlink, invlink};

%b = glmfit(XTrain,yTrain,'binomial', 'link', F);

%yTrainPred = glmval(b, XTrain, F)

%yTrainPred(yTrainPred >= 0.5) = 1;
%yTrainPred(yTrainPred < 0.5) = 0;
%accuracy_training = mean(double(yTrainPred == yTrain))*100;

%Plotting confusion matrix
%confMat = confusionmat(yTrain(1:end), yTrainPred);
%confusionchart(confMat)

% Calculating Precision, Recall, F1 Score
%for i =1:size(confMat,1) %https://uk.mathworks.com/matlabcentral/answers/262033-how-to-calculate-recall-and-
precision
%    precision(i)=confMat(i,i)/sum(confMat(i,:));
%end
%precision=sum(precision)/size(confMat,1);

%for i =1:size(confMat,1)
%    recall(i)=confMat(i,i)/sum(confMat(:,i));
%end
%recall(isnan(recall))=[];

%F_score=2*recall*precision/(precision+recall); %%F_score=2*1/((1/Precision)+(1/Recall));
```

Naïve Bayes for loop which was not useful due to the fact that we are only allowed mvnm with this type of dataset:

```
%% For loop to

DistributionNames = {'mvmn', 'mvmn'};
%width = [30 90]
%Kernel = ["normal" "triangle" "box" "epanechnikov"]
prior = {'empirical', 'uniform'}

for i=1:length(DistributionNames)
    for j = i:length(prior)

            rng default
            mdlNBl = fitcnb(XTrain, yTrain,'Prior', char(prior(j)), 'DistributionNames', ...
                char(DistributionNames(i)));
            %mdlNBl = fitcnb(XTrain, yTrain, 'ClassNames', {'yes', 'no'}, 'CrossVal', 'on',...
            %    'Prior', char(prior(j)), 'DistributionNames', char(DistributionNames(i)));
            %losses(i,j) = koldloss(mdlNBl);
            [pred, score, loss] = predict(mdlNBl, XTrain);
            %[pred,score] = resubPredict(mdlNBl);
            errNB = resubLoss(mdlNBl);
            confMat = confusionmat(yTrain.y, pred);
            for i =1:size(confMat,1) %https://uk.mathworks.com/matlabcentral/answers/262033-how-to-calculate-
recall-and-precision
                precision(i)=confMat(i,i)/sum(confMat(i,:));
            end
         precision=sum(precision)/size(confMat,1);

         for i =1:size(confMat,1)
                recall(i)=confMat(i,i)/sum(confMat(:,i));
         end
            recall(isnan(recall))=[];

         F_score=2*recall*precision/(precision+recall); %%F_score=2*1/((1/Precision)+(1/Recall));
         %
         accuracy = (confMat(1,1) + confMat(2,2)) / sum(sum(confMat));


    end
 end
figure('pos',[1000 1000 500 400])
```

```
confusionchart(confMat,{'yes', 'no'})

[X,Y,T,AUC] = perfcurve(yTrain.y, abs(score(:,2)),'yes');
figure('pos',[1000 1000 500 400])
plot(X,Y)
xlabel('False positive rate')
ylabel('True positive rate')
legend('Naive Bayes','Location','Best')
title('ROC for Classification by Naive Bayes')
```

Implementation details

The dataset was pretty noisy as there are many outliers in the numeric values. These were not removed to reflect the real world nature of this study. We did most of the data wrangling in python and the script is part of this submission. There was no missing data.

There was a decision to be made about working with matrices vs tables. In the end I opted for tables as this suited our models best and set the scene for fair comparison. Only for the lasso regularisation I used matrices due to acceptable input arguments for the lassoglm() function.

Glossary:

- *NB:* Naïve Bayes
- *LR*: Logistic Regression
- *Confusion Matrix*: table of errors counting true positives (TP), true negatives(TN), false positive(FP), false negatives(FN).
- *ROC*: Receiver Operating Characteristic curve is a plot that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied.
- *AUC*: Area Under Curve is used to summarise the quality of a ROC curve. The higher the better. Maximum is 1.
- *Precision*: TP/TP+FP
- *Recall*: TP/TP+FN
- *F-score*: 2(precision*recall/precision+recall) – the higher the better.
- *Accuracy*: ratio of correct predictions to false predictions.
- *GLM*: Generalised Liner Model of which Logistic Regression is an example.

Predictor descriptions:

1 - age (numeric)

2 - job : type of job (categorical:
"admin.","unknown","unemployed","management","housemaid","entrepreneur","student",
"blue-collar","self-employed","retired","technician","services")

3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

# related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: "unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

# other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)
16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

Dataset information from UCI:

Citation Request:
This dataset is public available for research. The details are described in [Moro et al., 2011].
Please include this citation if you plan to use this database:

[Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology.
In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS.

Available at: [pdf] http://hdl.handle.net/1822/14838
[bib] http://www3.dsi.uminho.pt/pcortez/bib/2011-esm-1.txt

1. Title: Bank Marketing
2. Sources
Created by: Paulo Cortez (Univ. Minho) and Sérgio Moro (ISCTE-IUL) @ 2012
3. Past Usage:
The full dataset was described and analyzed in:
S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology.
In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães,
Portugal, October, 2011. EUROSIS.
4. Relevant Information:
The data is related with direct marketing campaigns of a Portuguese banking institution.
The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required,
in order to access if the product (bank term deposit) would be (or not) subscribed.
There are two datasets:
1) bank-full.csv with all examples, ordered by date (from May 2008 to November 2010).
2) bank.csv with 10% of the examples (4521), randomly selected from bank-full.csv.
The smallest dataset is provided to test more computationally demanding machine learning algorithms (e.g. SVM).
The classification goal is to predict if the client will subscribe a term deposit (variable y).
5. Number of Instances: 45211 for bank-full.csv (4521 for bank.csv)
6. Number of Attributes: 16 + output attribute.
7. Attribute information:
For more information, read [Moro et al., 2011].