Imperial College Business School: Machine Learning in Finance **(2h)**

Exam Solution

Problem A: Credit Risk prediction

(Preprocessing)

(1) Determine which of these features are categorical and describe the one hot encoding preprocessing method.

- The categorical features are: "Geography", "Sex", "Purpose", "Education".
- The one hot encoding preprocessing method consists in representing each categorical feature (with K possible categories) by a vector of size K, with 0 everywhere and 1 in the position of the category.
- For instance, if the possible categories are A, B, C, the category B will be represented by the 3 dimensional vector $[0, 1, 0]$

(2) What transformation should be applied to the numerical values to normalize them.

The numerical features should be scaled. There are two ways of doing it.

- Standard scaling: By applying the following transformation, we redistribute the feature X with mean of zero and standard deviation of 1.

$$X \leftarrow \frac{X - \text{mean}(X)}{\text{std(X)}}$$

- Min-Max scaling: By applying the following transformation to make the data in the $[0, 1]$ range.

$$X \leftarrow \frac{X - \min(X)}{\max(X) - \min(X)}$$

(Logistic Regression)

(3) According to the previous results, which threshold should we use? Detail your answer

- T1 and T2 have the same FPR but T1 has a better TPR. So, T2 is preferred to T1.
- T2 and T3 share the same TPR but T2 has a better FPR. So, T2 is preferred to T3.
- As, T2 is preferred to both T1 and T3, we conclude that we should use the threshold T2.

(4) Which model looks preferable?

According to the "Gini" measure, the kernel logit looks preferable.

(5) Given the retained threshold, which model should be used?

Although the quadratic logit is not the optimal one in terms of "Gini coefficient", missing a "true defaulter" is much more important to a firm. So, we will prefer the model with the lowest type 1 errors, which is the quadratic logit.

(Decision Tree Algorithm)

(6) What will a null entropy value correspond to on such a graph?

For each node, a null entropy means that all the samples corresponding to this node belong to the same class.

(7) In the decision graph, which of the entropies e_i , with i from 1 to 9, will be equal to zero?

The null entropies correspond to the nodes where all the samples belong to the same class.
Hence:

$$e_2 = e_4 = e_6 = e_9 = 0$$

(8) From the decision graph, deduce the confusion matrix. Justify your answer.

Let's compute the TP, TN, FP, FN for each split from the root of the decision graph.

- From the first split, the left child gives us 150 TP.
- From the second split, the right child gives us 100 TN.
- From the third split, the left child gives us 100 TP.
- From the last split, the left child gives us 100 TN and the right child gives us 250 TP and 50 FP.

Which gives us the following confusion matrix.

	1	0
1	500	0
0	50	200

(9) Explain why the precision is a good evaluation metric.

There are two kinds of errors.

- The False Positive error, which consists in predicing a non default target for a defaulter.
- The False Negative error, which consists in predicting a default target for non defaulter.

As the False Positive error is considered as more dangerous than the False Negative error, we prefer the evaluation metric which focuses on the False Positive error (i.e the precision metric).

(10) Compare the precisions of the Decision Tree Algorithm and of the logistic regression associated to the T2 threshold (with TPR = 0.8 and FPR = 0.2) on the same training data.

The precision is expressed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- For the Decision Tree Algorithm:

$$\text{Precision} = \frac{500}{500 + 50} = 0.90$$

- For the Logistic Regression Model, we know that TPR = 0.8 and FPR = 0.6
  - Let's find TP:
    * We have:
    $$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
    * So,
    $$\text{TP} = \text{TPR} * (\text{TP} + \text{FN})$$
    * From the confusion matrix of the Decision Tree Algorithm, we can deduce the total number of samples associated to the target 1, which is 500. (i.e, TP + FN = 500).
    * Hence,
    $$\text{TP} = 0.8 * 500 = 400$$
  - Let's find FP:
    * We have:
    $$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$
    * So,
    $$\text{FP} = \text{FPR} * (\text{FP} + \text{TN})$$
    * From the confusion matrix of the Decision Tree Algorithm, we can deduce the total number of samples associated to the target 0, which is 250. (i.e, FP + TN = 250).
    * Hence,
    $$\text{FP} = 0.6 * 250 = 150$$
  - We can then deduce the precision of the Logistic Regression model:

$$\text{Precision} = \frac{400}{400 + 150} = 0.72$$

As a result, the Decision Tree algorithm has a better precision than the Logistic Regression Model.

(11) What should be the value of "Purpose" to make the DT algorithm predict a 'Good' target?

Let x^* be the new sample, we will go from the root of the decision graph to predict the target of x^*.

- As the credit value is $35k\$ < 50k\$$, x^* will be associated to the right child node of the first split.
- As the "education" value is "skilled", x^* will be associated to the left child node of the second split.
- Then, the "Duration" value is $11 < 15$, so x^* will be associated to the right child node of the third split.

- The target value will depend on the last split. If the value of ”Purpose” is ”P1”, x^* will be associated to the target 0. If, on the contrary, the value of ”Puropose” is ”P2”, ”P3” or ”P3”, the new sample x^* will be associated to the target 1.

Therefore, any value of ”Purpose” other than ”P1” will end up with a ”Good” target.

Problem B: A Simple Markov Model

(12) What method would you use to learn the parameters of a Markov Model on this small dataset

We would use the maximum likelihood estimation method.

(13) What is the optimal initial state vector π ?

Our training sequences start one time with states 0, 5 and 6, we get the following π parameter after the Maximum Likelihood Estimation:

$$\pi = \begin{pmatrix} 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 1/3 \\ 0 \\ 0 \end{pmatrix}$$

(14) What is the probability of transitioning from state 5 to state 1?

The probability of transitioning from state 5 to state 1 is the number of times we transition from state 5 to state 1, devided by the sum of all the transitions from state 5 to any state. Hence, the probability of transitioning from state 5 to state 1 is 1.

Problem C: Sentiment Analysis on Financial News

(C.1 A Deep Neural Network)

(15) What would be the shape of the tensor data after the preprocessing steps ?

The training dataset contains N sequences, each sequence is going to be encoded into a V-dimensional vector. The shape of the tensor training data is (N, V).

(16) Which activation functions should we use?

We can choose several activation functions for the three hidden layers (among ”sigmoid”, ”Relu”, ”tanh” , etc), but we should use the sigmoid activation function for the output layer as we are performing a binary classification task.

(17) Deduce the loss function associated with this classification problem.

As we are dealing with a binary classification problem, the correct loss function is the **binary cross entropy** loss function.

(18) Describe the problem highlighted by the validation loss?

The validation loss starts increasing from epoch 5. The model is no longer learning useful patterns for the training data. We are facing an overfitting problem.

(19) How would you solve it?

There are several ways to solve the overfitting problem. We can:

- Add more samples if it's possible.
- Reduce the complexity of the model by reducing the number of layers and the number of neurons per layer.
- Using some regularization techniques like Dropout or penalizing the loss function by the $\mathcal{L}^1$ norm or the $\mathcal{L}^2$ norm of the weights.

(20) Explain why the previous model will assign the same prediction to the sentences “What a great news in such a bad period” and “What a bad news in such a great period”?

- The encoding of the different sentences was performed regardless of the sequential nature of the data.
- We used a bog of words model to classify our financial news. So, the two sentences “What a great news in such a bad period” and “What a bad news in such a great period” are encoded with the same V-dimensional vector.
- Hence, the forward propagation will give the same prediction.

(21) Describe the different steps of creating a generative classifier using HMMs to perform the same classification task

- First, we need to separate our dataset into the subdataset (D_1) containing the sequences associated to the target 1 and the subdataset (D_0) containing the sequences associated to the target 0.
- We train a first HMM on the subdataset D_1. Let's call the fitted model (M_1).
- We train a second HMM on the subdataset D_0. Let's call the fitted model (M_0).

- Using Bayes Rule we can classify a new sequence $w^1, \dots, w^T$ as follows:

$$\begin{aligned} p(Y=1|w^1,\dots,w^T) &= \frac{p(w^1,\dots,w^T|Y=1)p(Y=1)}{p(w^1,\dots,w^T)} \\ &= \frac{p(w^1,\dots,w^T|Y=1)p(Y=1)}{p(w^1,\dots,w^T|Y=1)p(Y=1)+p(w^1,\dots,w^T|Y=0)p(Y=0)} \end{aligned}$$

  - From the fitted model M_0, we can deduce $p(w^1,\dots,w^T|Y=0)$
  - From the fitted model M_1, we can deduce $p(w^1,\dots,w^T|Y=1)$
  - $p(Y=1)$ is juste the frequency of the samples associated to a positive target.

(C.2 A Sequential Neural Network)

(22) Describe such a model by specifying the different layers, the different hyperparameters for each layer and how the shape of the data is changing after each layer transformation.

A proposed model: The sequence of the following layers:

- The tensor of data is of shape (N, T). It contains the sequences of integers representing the words.
- The first layer is an **Embedding layer**. It will transform the (N, T) tensor into an (N, T, D) tensor. Each integer (representing a word) will be encoded into a D-dimensional vector. As we want to use pretrainded word vectors, we don't want the embedding matrix to be updated during the training process. So, we will load a pretrained embedding matrix (like the one trained on the Wikipedia data using the word2vec algorithm).
- We can then stack p **LSTM** layers. Each of the p LSTM layers will return the sequence of outputs. Hence, The (N, T, D) tensor will be transformed into an (N, T, d_1) tensor, then (N, T, d_2), until the output of the last LSTM layer: an (N, T, d_p) tensor.
- We can then use a last LSTM layer, which only returns the last output: a number in the range $[0, 1]$. The final tensor will be an (N, 1) tensor.
- We can then train the model by setting the loss function to the binary cross entropy loss.