Both following tasks closely relates to the assigned obligatory Homework #2. You will use data set 'Caravan', which is a part of ISLR package available in R. Your presentations should be helpful also for all other students who work on Homework #2.

Task 1 – Precision and Recall depending on the classification threshold

In this exercise you will focus on evaluation of a binary classification task. You will predict the value of binary attribute Purchase. First, split the 5,822 available development examples into training and test set, just the same way as in Homework #2, i.e. 1000 randomly selected examples for test, and the rest for training. Now, build a default Random Forest model (with default parameter setting) and with 300 trees, like

\[ RF.300 = \text{randomForest}(\text{Purchase} \sim ., \text{caravan.train}, \text{ntree} = 300) \]

a) Use your test set to evaluate the RF model and make a plot with three curves in one picture to show how three different performance measures depend on the classification (cut-off) threshold. The three curves will be precision (use red color), recall (black color), and F1 measure (blue color). There should be the cut-off on the x-axis, and the performance measures on the y-axis.

Technical hints:
- To deal with different cut-off values you need to use 'probability' type prediction, like
  \[ \text{prediction.test.prob} = \text{predict}(RF.300, \text{caravan.test}, \text{type} = \text{"prob"}) \]
- and then to use a particular threshold value \( t \), like
  \[ \text{prediction.test} = \text{factor} (\text{prediction.test.prob}[ ,2] >= t, \text{levels} = \text{c("FALSE", "TRUE")}) \]
- To plot the required curves you should compute confusion matrix for each cut-off value in a sequence like
  \[ \text{cutoff.seq} = \text{seq}(0, 1, \text{by}=0.001) \]
- To draw more curves into one plot in R, use first \text{plot()} function, and then \text{points()}.

b) Now use the same data and the same cut-off values to make a plot with precision and recall, how they depend on FPR. There should be FPR on the x-axis, and the other performance measures on the y-axis. Recall in black and precision in red color. The recall curve will be in fact the ROC curve.

c) Do the same plots for several different number of trees in the RF model.
Task 2 – Classifier precision and ROC

This exercise is more theoretical. Consider a binary classification task and the relationship between precision and ROC curve. Using a given test set, each point on the ROC corresponds to a particular cut-off setting and both FPR and TPR values were computed from a particular confusion matrix. To answer the following questions you should assume that you know the number of positive examples in the test set \( P \), and the number of negative examples \( N \) as well, so that \( P + N = |T| \), where \( |T| \) is the test set size.

a) First, as a special case consider a diagonal ROC, i.e. TPR = FPR. What is the precision at different points on this special ROC? Answer it for FPR values 0.25, 0.5, and 0.75. Can you generalize from these three points?

b) What would be the ROC shape if the (hypothetical) predictor had a fixed precision for different FPR values? Write a function in R that plots the ROC curve for a given fixed precision. The function should have three parameters, like

\[
\text{plot_fixed_precision_roc} = \text{function}(P, N, \text{precision}) \{ \ldots \}
\]

Then, try to express the shape of this special ROC also analytically.

c) Now you should be already able to compute precision for any given pair of FPR and TPR values. Given some particular \( P \) and \( N \), write a formula to compute precision for any given point at ROC.
Task 1 – Illustrations

Evaluation on development test set with different cut-off thresholds
mtry = default, ntree = 300

Evaluation on development test set -- ROC and precision
mtry = default, ntree = 300
Task 2 – Solution

Basic definitions and relationships between variables:

<table>
<thead>
<tr>
<th>Observed variable</th>
<th>Meaning (= the number of)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = TP + FN</td>
<td>positives in the test set</td>
</tr>
<tr>
<td>N = TN + FP</td>
<td>negatives in the test set</td>
</tr>
<tr>
<td>P + N =</td>
<td>T</td>
</tr>
<tr>
<td>TP</td>
<td>true positives</td>
</tr>
<tr>
<td>TN</td>
<td>true negatives</td>
</tr>
<tr>
<td>FP</td>
<td>false positives</td>
</tr>
<tr>
<td>FN</td>
<td>false negatives</td>
</tr>
</tbody>
</table>

Performance measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR = TP/P</td>
<td>false positive rate = recall = sensitivity</td>
</tr>
<tr>
<td>FPR = FP/N</td>
<td>false positive rate</td>
</tr>
<tr>
<td>TP/(TP + FP)</td>
<td>precision</td>
</tr>
</tbody>
</table>

a) If TPR = FPR, then FP = TP * N/P. Thus precision = 1/(1 + N/P) = P/(P + N), which means that precision is constant at any point of the diagonal ROC.

Obviously, this poor level of precision would be reached even if the set of examples predicted as positives is selected as a random subset of the test set.

b) We have TP = TPR * P and FP = FPR * N, and thus

\[
\text{precision} = \frac{TP}{TP + FP} = \frac{TPR \cdot P}{TPR \cdot P + FPR \cdot N}.
\]

Hence

\[
\text{TPR} = \frac{\text{precision}/(1 - \text{precision}) \cdot N/P}{\text{FPR}},
\]

which is a linear function if precision is a fixed constant.

Since TPR is always \( \leq 1 \), the maximum value of FPR is

\[
\text{FPR}_{\text{max}} = \min(1, P/N \cdot (1 - \text{precision})/\text{precision}).
\]

```r
plot_fixed_precision_roc = function(P, N, precision){
  FPR_max = min(1, P/N * (1 - precision)/precision)
  TPR_max = min(1, precision/(1 - precision) * N/P)
  plot( c(0, FPR_max), c(0, TPR_max), type="l", col="red", lwd=3,
        xlim=c(0,1), ylim=c(0,1),
        xlab="FPR", ylab="TPR",
        main=paste0("ROC with fixed precision = ",
                     round(100*precision,1), "%\n",
                     "P = ", P, ", N = ", N)
    )
}
```

c) As derived above, precision = TPR * P/(TPR * P + FPR * N).
Task 2 – Illustrations

> plot_fixed_precision_roc(100,400,0.2)
> plot_fixed_precision_roc(100,500,0.2)

ROC with fixed precision = 20%
P = 100, N = 400

ROC with fixed precision = 20%
P = 100, N = 500

> plot_fixed_precision_roc(100,500,0.6)
> plot_fixed_precision_roc(500,250,0.5)

ROC with fixed precision = 60%
P = 100, N = 500

ROC with fixed precision = 50%
P = 500, N = 250