UNIT–6 : Performance Evaluation in R (PYQ Solution)

PART–A Answers (≈70 Words Each)

Q1. Define MSE and RMSE. (Expected)

MSE (Mean Squared Error) is the average of the squared differences between actual and predicted values: $MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$.

RMSE (Root Mean Squared Error) is its square root: $RMSE = \sqrt{MSE}$. RMSE is in the same unit as the target, so it is easier to interpret. Both penalise large errors heavily; lower is better. They are used for regression.

Q2. Define sensitivity and specificity. (Expected)

Sensitivity (recall, true positive rate) is the fraction of actual positives that the model correctly identifies: $TP/(TP + FN)$.

Specificity (true negative rate) is the fraction of actual negatives correctly identified: $TN/(TN + FP)$.

In medical tests, high sensitivity means few sick patients are missed, and high specificity means few healthy people are wrongly flagged.

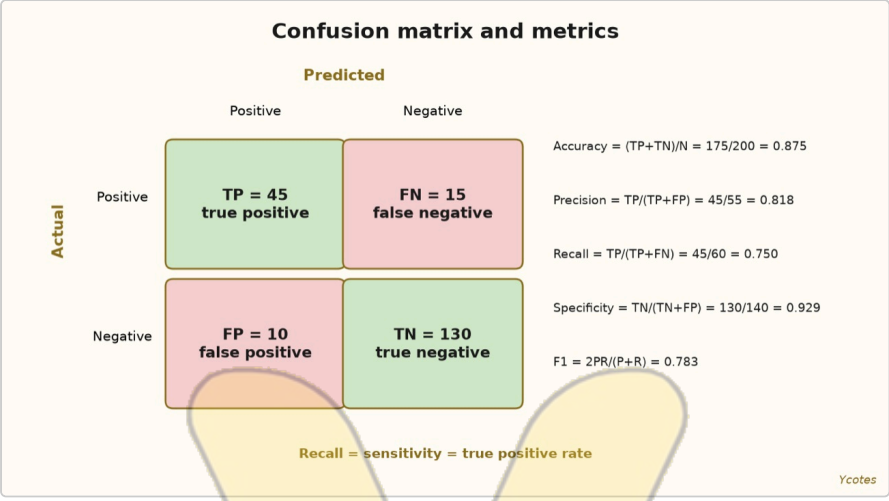
Q3. Define precision and recall. (Expected)

Precision is the fraction of predicted positives that are really positive: $TP/(TP + FP)$. It tells how trustworthy a positive prediction is.

Recall is the fraction of real positives that the model finds: $TP/(TP + FN)$. It tells how complete the detection is. The two usually trade off, and the F1 score $= 2PR/(P + R)$ combines them.

PART–B Answers (≈250 Words Each)

Q1. From the confusion matrix TP = 45, FP = 10, FN = 15, TN = 130 calculate accuracy, precision, recall and F1 score. (Expected)



	Predicted positive	Predicted negative
Actual positive	TP = 45	FN = 15
Actual negative	FP = 10	TN = 130

Total $N = 45 + 10 + 15 + 130 = 200$.

- **Accuracy** $= \frac{TP+TN}{N} = \frac{45+130}{200} = \frac{175}{200} = 0.875$ (87.5 percent)
- **Precision** $= \frac{TP}{TP+FP} = \frac{45}{55} = 0.818$
- **Recall** $= \frac{TP}{TP+FN} = \frac{45}{60} = 0.750$
- **F1** $= \frac{2 \times 0.818 \times 0.750}{0.818+0.750} = \frac{1.227}{1.568} = 0.783$

Also: specificity $= 130/140 = 0.929$ and false positive rate $= 10/140 = 0.071$.

```
library(caret)
confusionMatrix(factor(pred), factor(actual), positive = "1")
```

Interpretation: the model is accurate overall, but it misses 25 percent of the real positives, so recall is its weak point.

Q2. Calculate MSE and RMSE for given actual and predicted values.

(Expected)

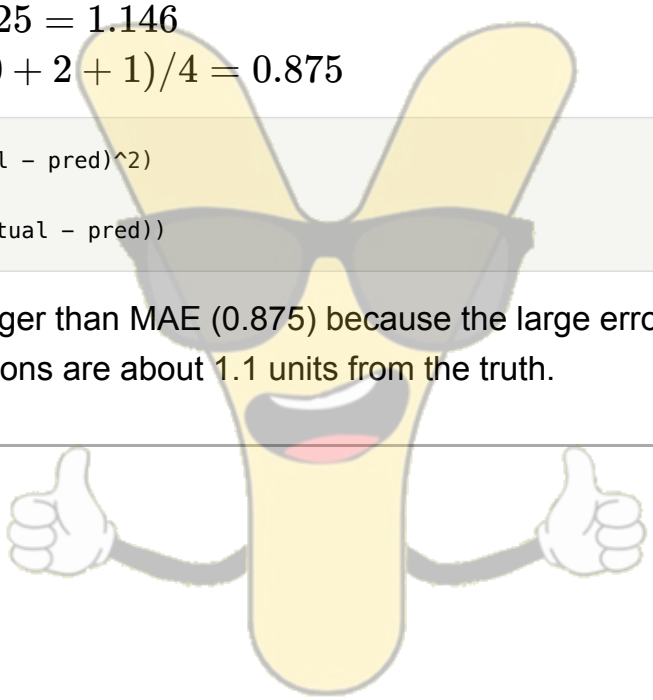
Data: actual y = 3, 5, 2, 7; predicted $\hat{y}$ = 2.5, 5, 4, 8.

y	predicted	error	squared error
3	2.5	0.5	0.25
5	5	0	0
2	4	-2	4
7	8	-1	1
Sum			5.25

$MSE = 5.25/4 = 1.3125$
$RMSE = \sqrt{1.3125} = 1.146$
$MAE = (0.5 + 0 + 2 + 1)/4 = 0.875$

```
mse  <- mean((actual - pred)^2)
rmse <- sqrt(mse)
mae  <- mean(abs(actual - pred))
```

RMSE (1.146) is larger than MAE (0.875) because the large error of 2 is squared. On average the predictions are about 1.1 units from the truth.



PART–C Answers (≈500 Words Each)

Q1. Explain binary classification and the confusion matrix. Explain all evaluation metrics and the ROC curve with R code. (Expected)

Binary classification predicts one of two classes, such as spam or not spam, fraud or genuine. Models such as logistic regression output a probability, and a **threshold** (default 0.5) converts it into a class label.

Confusion matrix:

	Predicted +	**Predicted -**
Actual +	TP (true positive)	FN (false negative)
Actual -	FP (false positive)	TN (true negative)

Metrics:

Metric	**Formula**	**Meaning**
Accuracy	$(TP + TN)/N$	Overall correctness
Precision	$TP/(TP + FP)$	Correctness of positive predictions
Recall / sensitivity	$TP/(TP + FN)$	Share of positives found
Specificity	$TN/(TN + FP)$	Share of negatives found
F1 score	$2PR/(P + R)$	Balance of precision and recall
FPR	$FP/(FP + TN) = 1 -$ specificity	False alarm rate

Worked values (TP 45, FP 10, FN 15, TN 130): accuracy 0.875, precision 0.818, recall 0.75, specificity 0.929, F1 0.783.

ROC curve: [img pds_roc]
It plots **TPR (recall)** on the y-axis against **FPR** on the x-axis for every possible threshold. A curve close to the top-left corner is better; the diagonal is random guessing. **AUC** is the area under the curve: 0.5 for random, 1.0 for perfect, and above 0.8 is usually good. A lower threshold raises both TPR and FPR.

```
library(caret); library(pROC)
model <- glm(y ~ ., data = train, family = binomial)
prob  <- predict(model, test, type = "response")
pred  <- factor(ifelse(prob > 0.5, 1, 0), levels = c(0, 1))
confusionMatrix(pred, factor(test[["y"]], levels = c(0, 1)), positive = "1")
r <- roc(test[["y"]], prob)
plot(r); auc(r)
```

Regression metrics for comparison: MSE, RMSE, MAE and R^2.

Conclusion: report several metrics together because each shows one side of the model.

Q2. Why is accuracy misleading on imbalanced data? Explain precision-recall trade-off and which metric to choose in which problem. (Expected)

Accuracy paradox: consider 1000 patients of whom only 10 have a disease. A useless model that predicts "healthy" for everyone gets $990/1000 = 99\%$ accuracy, yet its recall is $0/10 = 0$ and it finds no patient. Accuracy counts all correct answers equally, so the majority class hides the failure on the rare class.

Precision-recall trade-off: the threshold controls both.

- **Lower threshold:** more cases are called positive, recall rises, precision falls (more false alarms).
- **Higher threshold:** fewer positives are called, precision rises, recall falls (more misses).

Example with the matrix TP 45, FP 10, FN 15: precision 0.818 and recall 0.75. If the threshold is lowered and FN becomes 5 while FP becomes 30 (TP 55), precision $= 55/85 = 0.647$ and recall $= 55/60 = 0.917$.

Choosing the metric:

Problem	Costly error	Choose
Cancer or fraud detection	Missing a positive (FN)	**Recall** (sensitivity)
Spam filter, recommendation	Wrongly flagging (FP)	**Precision**
Imbalanced data, both errors matter	Both	**F1 score**
Comparing models across thresholds	Threshold choice	**AUC-ROC** or precision-recall AUC
Regression (price, demand)	Size of error	**RMSE**, MAE, R^2
Balanced classes	Any	Accuracy is acceptable

Good practice: balance the training data or use class weights, use **stratified k-fold cross-validation**, and look at the confusion matrix, not only a single number.

```
library(caret)
ctrl <- trainControl(method = "cv", number = 5, classProbs = TRUE, summaryFunction =
twoClassSummary)
train(Class ~ ., data = df, method = "glm", metric = "ROC", trControl = ctrl)
```

Conclusion: on imbalanced data prefer recall, precision, F1 or AUC to accuracy, and choose according to the cost of each type of error.
