Module 6: Basics of Predictive Performance Evaluation

Learning outcomes

1. Evaluate regression models using key performance metrics, including mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE).
2. Implement performance evaluation metrics to compare and select models for regression tasks.
3. Evaluate classification models using the confusion matrix and key performance measures.
4. Build a confusion matrix in Python to evaluate classification problems.
5. Apply evaluation methods to evaluate predictive performance in regression, classification and ranking problems.
6. Apply evaluation methods to measure predictive performance in regression, classification and ranking problems.
7. Create an execution plan outlining time management strategies, collaboration and model improvement approaches as well as risk management methods for the capstone competition project.

Regression problems

- They predict a continuous numeric output (e.g. temperature or house price).
- They measure raw values (such as individual errors).
- Their metrics are summary values for comparing models.

Common performance measures for regression problems

Metric	Description	Formula
Error (residual)	The most basic measure is the error for a single prediction. It's the difference between the actual value (y_i) and the predicted value ($\hat{y}_i$) for the $i-th$ data point.	$e_i = y_i - \hat{y}_i$
Sum of squared errors (SSE)	SSE is the sum of the squares of each error. It's used to address the issue of	$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

	error cancellation and to penalise larger errors.	
Sum of absolute errors (SAE)	SAE is the sum of the absolute difference of the errors.	$SAE = \|(y_i - \hat{y}_i)\|$
Mean error (ME)	ME is the average of the raw errors (they can cancel out).	$ME = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$

Common performance metrics for regression problems

Metric	Description	Formula
Mean absolute error (MAE)	MAE is the average absolute difference between the predictions and the actual values.	$MAE = \frac{1}{n}\sum_{i=1}^{n}\|(y_i - \hat{y}_i)\|$
Mean squared error (MSE)	MSE is the average of the squared differences.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Root mean squared error (RMSE)	RMSE is the square root of the MSE (interpretable in the original units).	$RMSE = \sqrt{MSE}$

Note: consider using RMSE when large errors are costly and using MAE when you want a balanced view.

Classification problems

- Predict categories or labels (e.g. spam vs not spam).

Confusion matrix

	Predicted positive	Predicted negative
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

Key metrics for regression problems

Metric	Formula	Use case
Accuracy	$\frac{(TP + TN)}{TP + TF + FP + FN}$	Overall correctness
Sensitivity (recall)	$\frac{TP}{(TP + FN)}$	Important when missing positives is risky (e.g. disease detection)
Specificity	$\frac{TN}{(TN + FP)}$	Important when false alarms are costly (e.g. spam filters)
Precision	$\frac{TP}{(TP + FP)}$	Focuses on the correctness of positive predictions

Note: always consider context when choosing which metric to optimise.

Ranking problems and lift charts

- A ranking problem is used when prioritising predictions matters more than exact accuracy (e.g. fraud detection or top recommendations).
- A lift chart is a visual tool to show how well the model prioritises the most important cases compared with random selection.
    - Key elements:
        - The x-axis represents the proportion or number of cases considered, ordered by the model's predicted score or rank

(from most to least likely positive).
    - The y-axis shows the cumulative gain or cumulative sum of the target metric (such as positive responses, ranks or scores).
  - Lines in the chart:
    - The model line is the cumulative gain when selecting cases according to the model's ranking.
    - The perfect model line is the ideal cumulative gain if the model perfectly ranks all positives first.
    - The random baseline line is the expected cumulative gain from random selection.

Note: the more your model line pulls away from the random line (upwards), the better your model is at ranking the top outcomes.

Implementation in Python

- For classification tasks, `scikit-learn` provides the function `confusion matrix` in the `sklearn.metrics` module. This function takes two arrays/lists, i.e. the true labels and the predicted labels and returns a confusion matrix.

```
import pandas as pd
 from sklearn.metrics import confusion_matrix
 # Identify the true labels and predicted labels
y_true = df['actual_status']

y_pred = df['predicted_status']

# Create the confusion matrix
cm = confusion_matrix(y_true, y_pred)

print("Confusion Matrix:")

print(cm)
```

- For lift charts:
  - Prepare data and define ranking scenarios.
  - Calculate the cumulative target variable.
  - Plot the lift chart.

Note: overall, choose evaluation metrics based on the following.

- The type of problem (regression, classification, ranking)
- The impact of different kinds of errors
- What matters more – accuracy, recall, precision or prioritisation