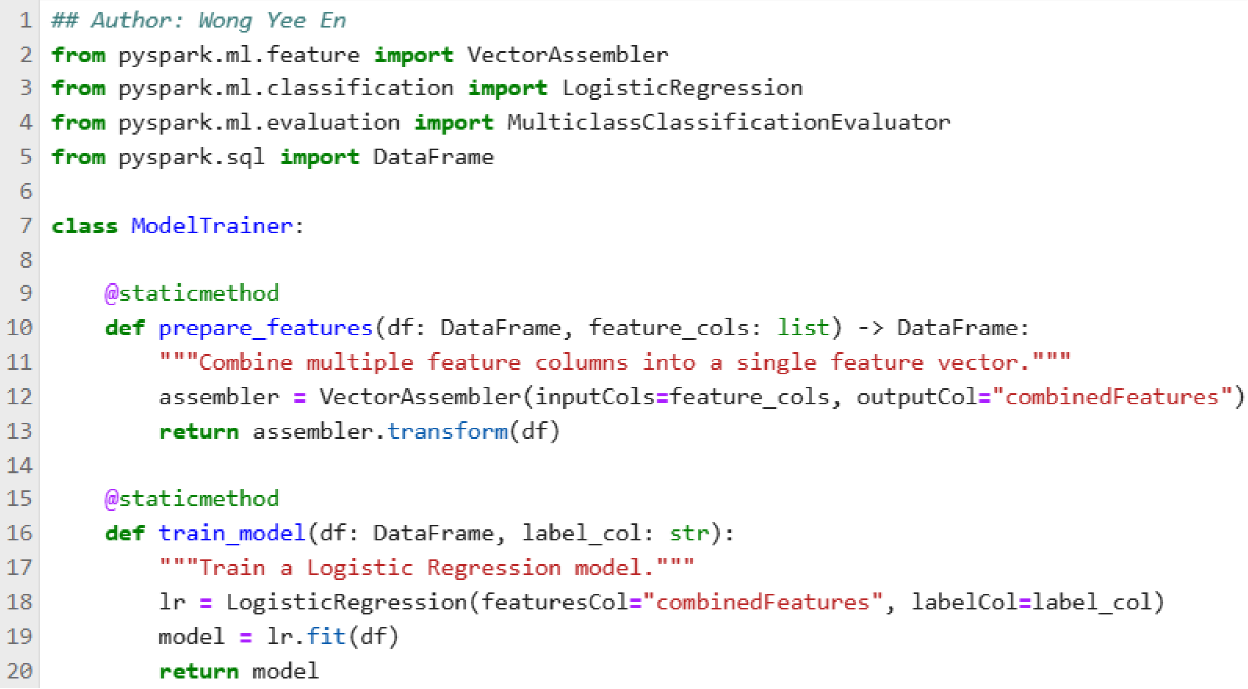
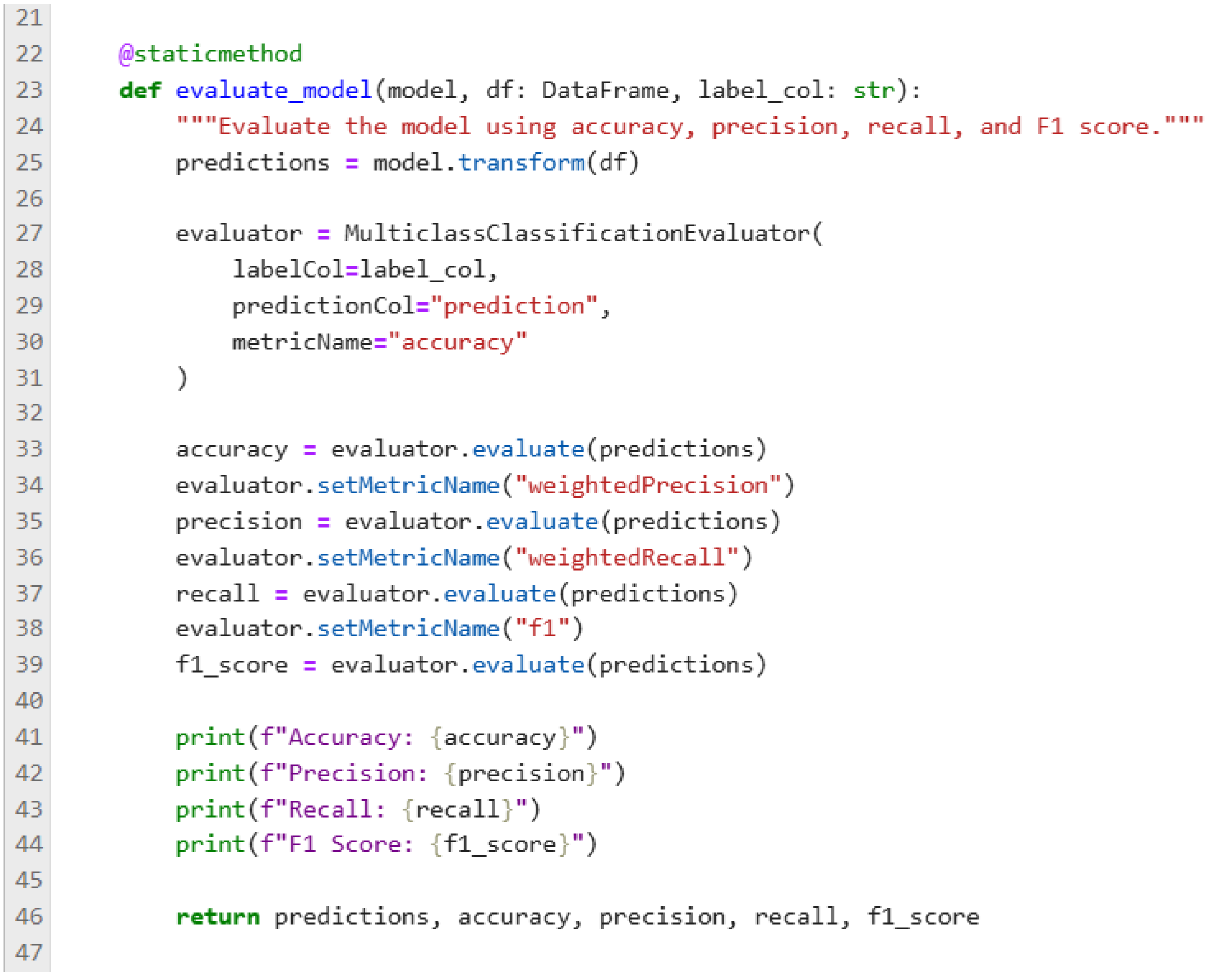


```python
## Author: Wong Yee En
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.sql import DataFrame

class ModelTrainer:

    @staticmethod
    def prepare_features(df: DataFrame, feature_cols: list) -> DataFrame:
        """Combine multiple feature columns into a single feature vector."""
        assembler = VectorAssembler(inputCols=feature_cols, outputCol="combinedFeatures")
        return assembler.transform(df)

    @staticmethod
    def train_model(df: DataFrame, label_col: str):
        """Train a Logistic Regression model."""
        lr = LogisticRegression(featuresCol="combinedFeatures", labelCol=label_col)
        model = lr.fit(df)
        return model
```



```python

    @staticmethod
    def evaluate_model(model, df: DataFrame, label_col: str):
        """Evaluate the model using accuracy, precision, recall, and F1 score."""
        predictions = model.transform(df)

        evaluator = MulticlassClassificationEvaluator(
            labelCol=label_col,
            predictionCol="prediction",
            metricName="accuracy"
        )

        accuracy = evaluator.evaluate(predictions)
        evaluator.setMetricName("weightedPrecision")
        precision = evaluator.evaluate(predictions)
        evaluator.setMetricName("weightedRecall")
        recall = evaluator.evaluate(predictions)
        evaluator.setMetricName("f1")
        f1_score = evaluator.evaluate(predictions)

        print(f"Accuracy: {accuracy}")
        print(f"Precision: {precision}")
        print(f"Recall: {recall}")
        print(f"F1 Score: {f1_score}")

        return predictions, accuracy, precision, recall, f1_score

```