 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Machine learning(01CT0519)	Aim: Implement Regression as Classification problem	
Practical_Assignment 1_ Linear Regression	Date:	Enrolment No: 92301733051

```

✓ [31] import pandas as pd
0s      import numpy as np
      from sklearn.preprocessing import OneHotEncoder

✓ [32] df = pd.DataFrame({
0s      'Student_ID': [1, 2, 3, 4, 5],
      'CGPA': [7.90, 7.39, 8.02, 8.72, 7.31],
      'Internships': [3, 0, 2, 4, 2],
      'Placed': ['Yes', 'Yes', 'Yes', 'Yes', 'Yes'],
      'Salary (INR LPA)': [17.63, 28.37, 8.95, 22.59, 19.67]
    })

✓ [33] bins = [0, 10, 20, 30]
0s      labels = ['Low', 'Medium', 'High']
      df['Salary_Class'] = pd.cut(df['Salary (INR LPA)'], bins=bins, labels=labels)

✓ [34] x = df[['CGPA', 'Internships']].values # 2 features
0s      y = df['Salary_Class'].values.reshape(-1, 1)

✓ [36] # One-hot encoding for classes
0s      from sklearn.preprocessing import OneHotEncoder

      encoder = OneHotEncoder(sparse_output=False) # updated parameter name
      Y = encoder.fit_transform(y)

✓ [37] X_bias = np.c_[np.ones(X.shape[0]), X]
0s

+ Code + Text

✓ [38] W = np.linalg.pinv(X_bias).dot(Y)
0s

✓ [39] scores = X_bias.dot(W)
0s

✓ [40] probs = np.exp(scores) / np.sum(np.exp(scores), axis=1, keepdims=True)
0s

✓ [41] pred_class_idx = np.argmax(probs, axis=1)
0s


✓ [42] pred_classes = encoder.inverse_transform(np.eye(len(labels))[pred_class_idx])
0s

✓ [44] # Convert back to class labels
0s      pred_classes = encoder.inverse_transform(np.eye(len(labels))[pred_class_idx])

      # Flatten before adding to DataFrame
      df['Predicted_Class'] = pred_classes.ravel()

      print(df[['Student_ID', 'CGPA', 'Internships', 'Salary (INR LPA)', 'Salary_Class', 'Predicted_Class']])

```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Machine learning(01CT0519)	Aim: Implement Regression as Classification problem	
Practical_Assignment 1_ Linear Regression	Date:	Enrolment No: 92301733051

Student_ID	CGPA	Internships	Salary (INR LPA)	Salary_Class	\
0	1	7.90	3	17.63	Medium
1	2	7.39	0	28.37	High
2	3	8.02	2	8.95	Low
3	4	8.72	4	22.59	High
4	5	7.31	2	19.67	Medium

Predicted_Class
0 Medium
1 High
2 High
3 High
4 Medium

```

from sklearn.metrics import confusion_matrix, classification_report
y_true = np.array(y).ravel()
y_pred = pred_classes.ravel()

print("Confusion Matrix:\n", confusion_matrix(y_true, y_pred, labels=labels))
print("\nClassification Report:\n", classification_report(y_true, y_pred, labels=labels))

```

```

Confusion Matrix:
[[0 0 1]
 [0 2 0]
 [0 0 2]]

Classification Report:
              precision    recall  f1-score   support

   Low           0.00        0.00        0.00         1
   Medium        1.00        1.00        1.00         2
   High          0.67        1.00        0.80         2

 accuracy          0.80
 macro avg         0.56        0.67        0.60         5
 weighted avg      0.67        0.80        0.72         5

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to nan. Please use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to nan. Please use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to nan. Please use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

```