COIMBATORE INSTITUTE OF TECHNOLOGY.

COIMBATORE.

Submitted by:

Nareshkumar G

1833029

Problem Statement 2:

1. Import the necessary libraries like pandas, Numpy, sns etc.
2. Reading the data with the help of pandas
3. Removing the column names of dataset so that it is easy to use
4. Separating the dataset into numeric dataset and non-numeric dataset
5. Summarising data
6. Doing EDA for our dataset, we found out the followings:

   Interpretation:

   - There are 4 numeric and 1 categorical column.
   - There are 400 rows and 5 columns.
   - Columns in the dataset are ['User_ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'] respectively.
   - Dataframe has 5 columns. It is clear that there is no missing value present.
   - There are 0 duplicate Rows in the dataset.
   - We can see that there is positive correlation between Age and purchased.
   - We can see that most of the people didn't purchased
   - We can see that person whose age is above 40 are more likely to purchase.
   - Female whose age is above 40 are more likely to purchase.
   - Average age for a person to purchase is 38.
   - It is clear that User_ID is just a set of unique values because its mean is 1. It is recommended to remove these types of columns.
7. Feature Engineering:
   - There is no missing values.
   - Checking if missing values are present then processed further.
   - Removing duplicates values if present.
   - Converting Categorical column to numeric column.
8. Feature Selection and spilt:
   - Splitting the training and testing set from dataset.
9. Feature Scaling:
   - Scaling the train and test data so that every column is in certain range rather than influencing each other.
   - By using StandardScaler(), we can scale the data.
10. Model fitting:
    - Here we use Logistic Regression because dependent variable(target) is discrete.
    - Model:
      - ➢ Output => 0 or 1: (<=50k or >50k)
      - ➢ Hypothesis => Z = WX + B
      - ➢ Activation function => Sigmoid (0,1)
      - ➢ Decision boundary => threshold = 0.5 (1 if y >0.5, 0 if y <0.5)
      - ➢ Cost function => Mean squared error (-y * log(h(x) - (1 - y) * log (1 - h(x))
      - ➢ Gradient descent => w = w – (learning rate* dw*T) & b = b – (learning rate* db)

        W – weight, b – bias.

- Gradient descent updates the weights if cost function converges (minimize). There will be global minimum that mean where ever the point gradient descent starts it always converges at same point.

11. Predict the class 1 or 0.
12. Root mean Square error is calculated and RMSE is 0.44
13. Confusion matrix is build based on test set and predicted test set. Classification report is generated. Accuracy is around 89%.
14. Decision tree model is used with criterion "entropy" from Sklearn.
15. Then we used listedColormap for output plot.
16. Conclusion: Models build from scratch and from Sklearn classifies the same.