CSE472 (Machine Learning Sessional)
Assignment 1: Data Preprocessing and Feature Engineering

Introduction:

Data processing cleans and prepares raw data, removing errors and inconsistencies. **Feature engineering** creates new, meaningful variables from the data to help the model learn better. These steps are important for improving accuracy and reliability in machine learning models.

In this assignment, you will learn the basic steps of importing a dataset into for a machine learning project, preprocessing the different variables of the dataset, removing missing values and redundancies, converting non-numerical variables into numeric format, normalization of the data ,correlation analysis of different variables with the targeted output, and selection of important features from the dataset for finally fitting a dataset into a machine learning pipeline.

Please note that you can use python library functions for this assignment.

Required Dataset:

In this assignment we will use the "IBM HR Analytics Employee Attrition & Performance" dataset. You can download the dataset from the following link:
https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Basic description about the dataset:

The dataset is used to understand the factors that contribute to employee attrition (indicating whether an employee has left the company (Yes/No)) and to build models that can predict which employees are likely to leave.
The dataset consists of 35 columns (features) and 1470 rows (employee records).

Personal Attributes: Age, Gender, Marital Status, Education, etc.
Job-Related Attributes: Job Role, Department, Job Level, Job Satisfaction, etc.
Performance Metrics: Years at Company, Years in Current Role, Performance Rating, etc.
Compensation: Monthly Income, Stock Option Level, etc.
Attrition: The target variable, indicating whether an employee has left the company (Yes/No).

With the given dataset, you need to perform certain tasks which are given below. Remember that these steps are always crucial in dealing with any machine learning project. So we expect you to understand all these steps and implement them yourselves.

Required Tasks:

a. Understanding the dataset:

1. Import the dataset in a notebook environment with python library : “Pandas”
2. Show the number of attributes (columns) and number of records (rows)
3. Show the statistics of the dataset (column wise mean, standard deviation, max, min etc)
4. Count the number of missing values in the dataset
5. Count the number of duplicate values in the dataset.

b. Data cleaning:

1. If you find any missing values in the dataset (nan values) replace those data with the column wise mean.
2. If you find any duplicates in the dataset, keep just one copy of the data.
3. Remember, if any row in the target column (Attrition) is missing, you must drop that row

c. Creation of input and output features:

1. You need to split the data into two parts. The “Features” variable will consist of all the columns in the dataset except the target column. And the “Labels” variable will contain only the column.

d. Conversion of features into numeric values:

1. You will notice that a number of columns in the dataset contains text (string type) features. For example, the target column also contains labels in the form : “Yes”/ “No”. You first need to convert these columns into numeric features.
2. For doing that, you need to first convert such columns which are not numeric types, into categorical types. Then you need to perform one hot encoding on that column, which will divide that column into multiple one hot type column. To better understand this approach, follow this link: get_dummies_in_pandas.
   (You can use library function here)
3. Remember that, you could have performed label encoding (Label Encoding) instead of one hot encoding. However, giving strict numeric values to some labels might create a bias. For example, if you convert a column ‘Department’ like this:
   Sales → 0
   Research and Development → 1
   Human Resource → 2
   This might create a bias in the model to give the human resource variable greater value. However, if there are only 2 different values in a column, you can perform label encoding instead of one hot. This will reduce the number of new columns.

e. Scaling of the features:

1. You will see in the dataset that, the ranges of values in a column varies significantly with the values range from a different column. This will surely hamper the training process of the ML algorithm. To resolve this issue, we apply scaling.
2. There are two types of scaling you can perform: StandardScaling and MinMaxScaling. You need to perform both type of scaling on the dataset and verify which works well (It is expected you do this in a functional way, so whenever you prefer one scaling over another, you can simply change your preference in the argument of this scaling function).
   Standard Scaling (Z-score normalization) transforms features so that they have a mean of 0 and a standard deviation of 1.
   Min-Max Scaling (also known as normalization) scales the data to a fixed range, typically between 0 and 1.
   (You can use library function here)
3. Remember, never scale the target variable. Only scale the feature variables. Please note that, you should avoid scaling the one hot type column that you get in d(2). They are already scaled between zero and one.

f. Correlation Analysis:

1. At this point you have “Features” that contain numeric features, and a target column. You now need to perform a correlation analysis on this processed dataset. You need to show the correlation of every column with the target column.
2. You need to show the output result (correlation of each variable with the target column) in the notebook. (You can use library function here)
3. You can also show a correlation matrix for visualization.
4. Select the top 20 columns that have the highest correlation with the target variable. For each of these columns, you can perform a 1D scatter plot to see how these variables help to understand the separation between the different classes of the target variable.

g. Validating the pipeline (Bonus Task) :

1. Now you have a cleaned “Features” data with 20 columns and also a target column.
2. Now you can use the following codebase to perform a classification with a simple machine learning model (You will eventually learn to implement these models on your own throughout this course, here we are providing a skeleton with python libraries just for a better understanding of the importance of these previous steps! No need to understand them all now.)

```python
# Import necessary libraries
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assume X and y are your input matrices
# X --> (number of rows, number of columns), already scaled
# y --> binary target class (0 or 1)

# Dummy example data (replace these with your actual data)
# X = np.random.rand(100, 5)  # Example feature matrix with 100 rows and 5 columns
# y = np.random.randint(0, 2, 100)  # Example binary target vector

# Step 1: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 2: Initialize the Logistic Regression classifier
clf = LogisticRegression()

# Step 3: Train the classifier on the training data
clf.fit(X_train, y_train)

# Step 4: Make predictions on the test set
y_pred = clf.predict(X_test)

# Step 5: Evaluate the classifier's performance
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of Logistic Regression classifier: {accuracy:.2f}")
```

Marking Criterion:

Steps	Points
Understanding the dataset	15
Data cleaning	10
Creation of input and output features	5
Conversion of features into numeric values	25
Scaling of the features	20
Correlation Analysis	25
Validating the pipeline	5 (Bonus)

Evaluation

1. Submission deadline **Friday 10:00 PM, September 6, 2024**.
2. You have to reproduce your experiments during in-lab evaluation. Keep everything ready to minimize delay.
3. You are likely to give online tasks during evaluation which will require you to modify your code.
4. You will be tested on your understanding through viva-voce.

Warning

1. Don't copy! We regularly use copy checkers. Do not copy codes from online resources and LLMs.
2. First time copier and copyee will receive negative marking because of dishonesty. Their default is bigger than those who will not submit.
3. Repeated occurrence will lead to severe departmental action and jeopardize your academic career. We expect fairness and honesty from you. Don't disappoint us!