JAYPEEUNIVERSITYOFINFORMATIONTECHNOLOGY
WAKNAGHAT, P.O.–WAKNAGHAT, TEHSIL– KANDAGHAT,
DISTRICT– SOLAN (H.P.) PIN–173234(INDIA)
(Established by H.P. State Legislature vide Act No. 14 of 2002)

JAYPEE EDUSPHERE

IGNITED MINDS
INSPIRED SOULS

18B1WCI674: MACHINE LEARNING LAB

Assignment-6 (EDA and Scikit-learn)

Jan 12, 2026

Instructions

1. Use Jupyter Notebook or Google Colab.
2. Load the dataset from a CSV file.
3. Perform Exploratory Data Analysis (EDA).
4. Apply data preprocessing using Scikit-learn.
5. Train and evaluate a machine learning model.
6. Write simple observations at each stage.

Overview

Exploratory Data Analysis (EDA) helps in understanding the dataset using statistics and visualizations.
Data preprocessing prepares the dataset for machine learning algorithms.
Scikit-learn provides simple tools for preprocessing, model training, and evaluation.

Dataset Loading and Basic Understanding

1. Load the dataset from CSV file:

```
df = pd.read_csv("dataset.csv")
```

2. Display the first 5 rows of the dataset:

   —

3. Display the last 5 rows of the dataset:

   —

4. Check the number of rows and columns:

   —

5. Display column names of the dataset:

   —

6. Check data types of each column:

   —

7. Check for missing values in the dataset:

   —

1. Exploratory Data Analysis (EDA)

1. Plot histogram of a numerical feature:

   —

2. Plot count plot for a categorical feature:

   —

3. Detect outliers using box plot:

   —

4. Plot correlation heatmap between numerical features:

   —

5. Identify important patterns and trends from plots:

   —

6. Write simple observations based on EDA:

   —

—

2. Data Preprocessing using Scikit-learn

1. Handle missing values using mean or mode:

```
df["Age"].fillna(df["Age"].mean(), inplace=True)
```

2. Encode categorical features:

```
from sklearn.preprocessing import LabelEncoder

df["Gender"] = LabelEncoder().fit_transform(df["Gender"])
```

3. Separate features and target variable:

```
X = df.drop("target", axis=1)
y = df["target"]
```

4. Feature scaling using StandardScaler:

```
from sklearn.preprocessing import StandardScaler
X = StandardScaler().fit_transform(X)
```

5. Split dataset into training and testing sets:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, random_state=42)
```

3. Machine Learning Model using Scikit-learn

1. Select a machine learning algorithm:
   - Logistic Regression
   - K-Nearest Neighbors

2. Train the model:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
```

3. Predict test data:

```
y_pred = model.predict(X_test)
```

4. Evaluate model performance:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

5. Display confusion matrix:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

6. Generate classification report:

```
from sklearn.metrics import classification_report
classification_report(y_test, y_pred)
```

4. Now. Solve for Regression