230962258_Aadiv_49_ML-3

August 5, 2025

Machine Learning Lab #3

DATA PREPROCESSING AND REGRESSION

[ ]:

```

```

[18]:

```
#Generic Imports

import numpy as np
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
from scipy.sparse import csr_matrix
import seaborn as sns
```

[19]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
```

Questions

1. Consider the hepatitis/ pima-indians-diabetes csv file, perform the following date preprocessing.

1. Load data in Pandas.

[20]:

```
df = pd.read_csv("/home/cvl-5aiml-a2/Documents/ML_230962258_AIML_A/Datasets/
  ↪diabetes_csv.csv")
```

2. Drop columns that aren't useful.

[21]:

```
# Dropped below:
df = df.drop("Pregnancies",axis=1)
df = df.drop("SkinThickness", axis = 1)
df.reset_index(drop=True, inplace=True)

df.head()
```

[21]:

```
   Glucose  BloodPressure  Insulin   BMI  DiabetesPedigreeFunction  Age  \
0      148             72        0  33.6                     0.627   50
1       85             66        0  26.6                     0.351   31
2      183             64        0  23.3                     0.672   32
```

```
3       89             66       94  28.1                     0.167   21
4      137             40      168  43.1                     2.288   33

   Outcome
0        1
1        0
2        1
3        0
4        1
```

3. Drop rows with missing values.

```
[22]: df.dropna(axis = 0)
```

```
[22]:      Glucose  BloodPressure  Insulin   BMI  DiabetesPedigreeFunction  Age  \
0        148             72        0  33.6                     0.627   50
1         85             66        0  26.6                     0.351   31
2        183             64        0  23.3                     0.672   32
3         89             66       94  28.1                     0.167   21
4        137             40      168  43.1                     2.288   33
..       …            …      …   …                       …  …
763      101             76      180  32.9                     0.171   63
764      122             70        0  36.8                     0.340   27
765      121             72      112  26.2                     0.245   30
766      126             60        0  30.1                     0.349   47
767       93             70        0  30.4                     0.315   23

     Outcome
0          1
1          0
2          1
3          0
4          1
..       …
763        0
764        0
765        0
766        1
767        0

[768 rows x 7 columns]
```

4. Create dummy variables

5. Take care of missing data.

6. Convert the data frame to NumPy.

```
[23]: np_df = df.to_numpy()
```

7. Divide the data set into training data and test data.

[24]:

```
X = df.drop('Outcome', axis=1)
Y = df['Outcome']

x_train = X[:80]
y_train = Y[:80]
x_test = X[80:]
y_test = Y[80:]
```

2. a. Construct a CSV file with the following attributes:

Study time in hours of ML lab course (x)

Score out of 10 (y)

The dataset should contain 10 rows.

[37]:

```
x = np.random.randint(0, 10, 10)
y = np.random.randint(0, 10, 10)
data = {'x': x, 'y': y}
df = pd.DataFrame(data)
df.to_csv('Q2a.csv', index=False)

print("CSV file 'Q2a.csv' has been created.")
```

```
CSV file 'Q2a.csv' has been created.
```

b. Create a regression model and display the following:

Coefficients: B0 (intercept) and B1 (slope)

RMSE (Root Mean Square Error)

Predicted responses

[38]:

```
df_read = pd.read_csv('Q2a.csv')
print("\nFirst 5 rows of the dataset:")
print(df_read.head())
print("\nInformation about the dataset:")
print(df_read.info())
```

```

First 5 rows of the dataset:
   x  y
0  7  1
1  5  7
2  9  5
3  2  9
4  5  6

Information about the dataset:
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       10 non-null     int64
 1   y       10 non-null     int64
dtypes: int64(2)
memory usage: 292.0 bytes
None
```

c. Create a scatter plot of the data points in red color and plot the graph of x vs. predicted y in blue color.

d. Implement the model using two methods:

Pedhazur formula (intuitive)

Calculus method (partial derivatives, refer to class notes)

[ ]:

```
df = pd.read_csv('Q2a.csv')
x = df['x'].values
y = df['y'].values
n = len(x)

# Method 1: Pedhazur Formula (Intuitive/Analytical)
x_mean = np.mean(x)
y_mean = np.mean(y)

# slope
numerator_b1 = np.sum((x - x_mean) * (y - y_mean))
denominator_b1 = np.sum((x - x_mean)**2)
b1_pedhazur = numerator_b1 / denominator_b1
#intercept
b0_pedhazur = y_mean - b1_pedhazur * x_mean

# ----------------------------------------
# Method 2: Calculus Method (Partial Derivatives)
# ----------------------------------------

sum_x = np.sum(x)
sum_y = np.sum(y)
sum_x2 = np.sum(x**2)
sum_xy = np.sum(x * y)

A = np.array([[n, sum_x], [sum_x, sum_x2]])
B = np.array([sum_y, sum_xy])

try:
    solution = np.linalg.solve(A, B)
    b0_calculus = solution[0]
    b1_calculus = solution[1]
except np.linalg.LinAlgError:
    b0_calculus, b1_calculus = None, None
    print("Error: Matrix is singular. Cannot solve the system of equations.")
```

e. Compare the coefficients obtained using both methods and compare them with the analytical solution.

[43]:

```
print("Coefficients from Pedhazur Formula:")
print(f"B0 (Intercept): {b0_pedhazur}")
print(f"B1 (Slope): {b1_pedhazur}")
print("\nCoefficients from Calculus Method:")
if b0_calculus is not None:
    print(f"B0 (Intercept): {b0_calculus}")
    print(f"B1 (Slope): {b1_calculus}")
```

```
Coefficients from Pedhazur Formula:
B0 (Intercept): 5.447619047619048
B1 (Slope): -0.30952380952380953

Coefficients from Calculus Method:
B0 (Intercept): 5.447619047619046
B1 (Slope): -0.3095238095238093
```

f. Test your model to predict the score obtained when the study time of a student is 10 hours.

[45]:

```
study_time_new = 10
predicted_score_new = b0_calculus + b1_calculus * study_time_new

print(f"\nPrediction:")
print(f"When the study time is {study_time_new} hours, the predicted score is␣
  ↪{predicted_score_new:2f}.")
```

```

Prediction:
When the study time is 10 hours, the predicted score is 2.352381.
```