

5 b. Linear Regression and Logistic Regression with the Diabetes Dataset Using Python Machine Learning

Aim

In this experiment we use the diabetes dataset from sklearn and then we need to implement the Linear Regression over this:

Procedure

Load sklearn Libraries.

Load Data

Load the diabetes dataset

Split Dataset

Creating Model Linear Regression and Logistic Regression

Make predictions using the testing set

Finding Coefficient And Mean Square Error

Program

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn import datasets, linear_model
```

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.model_selection import train_test_split
```

```
#To calculate accuracy measures and confusion matrix
```

```
from sklearn import metrics
```

```
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
```

```
diabetes_X = diabetes_X[:, np.newaxis, 2]
```

```
# Split the data into training/testing sets
```

```
diabetes_X_train = diabetes_X[:-20]
```

```
diabetes_X_test = diabetes_X[-20:]
```

```
# Split the targets into training/testing sets
```

```
diabetes_y_train = diabetes_y[:-20]
```

```
diabetes_y_test = diabetes_y[-20:]
```

```
# Create linear regression object
```

```
regr = linear_model.LinearRegression()
```

```
# Train the model using the training sets
```

```
regr.fit(diabetes_X_train, diabetes_y_train)
```

```
# Make predictions using the testing set
```

```

diabetes_y_pred = regr.predict(diabetes_X_test)

# Create Logistic regression object
Logistic_model = LogisticRegression()
Logistic_model.fit(diabetes_X_train, diabetes_y_train)

# The coefficients
print('Coefficients: \n', regr.coef_)
# The mean squared error
print('Mean squared error: %.2f'
      % mean_squared_error(diabetes_y_test, diabetes_y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f'
      % r2_score(diabetes_y_test, diabetes_y_pred))

y_predict = Logistic_model.predict(diabetes_X_train)
#print("Y predict/hat ", y_predict)
y_predict

```

Output

```

Coefficients:
[938.23786125]
Mean squared error: 2548.07
Coefficient of determination: 0.47

```