

School of Computer Science Engineering and Technology

Course- BTech
Course Code: CSET 301
Year- 2022-23
Date-

Type- Specialization Core
Course Name- AIML
Semester- 4th
Batch-

Lab Assignment W1L2

Logistic Regression Model for Divorce Prediction

Part 1.1: Implement linear regression from scratch

Logistic regression

Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as W) to predict an output value (y). A key difference from linear regression is that the output value being modelled is a binary values (0 or 1) rather than a continuous value.

Dataset

The dataset is available at "data/divorce.csv" in the respective challenge's repo.

Original Source: [UCI Machine Learning Repository: Divorce Predictors data set Data Set](#)

Dataset is based on rating for questionnaire

filled by people who already got divorce and those who is happily married.

Features (X)

1. Atr1 - If one of us apologizes when our discussion deteriorates, the discussion ends. (Numeric | Range: 0-4)
2. Atr2 - I know we can ignore our differences, even if things get hard sometimes. (Numeric | Range: 0-4)
3. Atr3 - When we need it, we can take our discussions with my spouse from the beginning and correct it. (Numeric | Range: 0-4)
4. Atr4 - When I discuss with my spouse, to contact him will eventually work. (Numeric | Range: 0-4)
5. Atr5 - The time I spent with my wife is special for us. (Numeric | Range: 0-4)
6. Atr6 - We don't have time at home as partners. (Numeric | Range: 0-4)
7. Atr7 - We are like two strangers who share the same environment at home rather than family. (Numeric | Range: 0-4)

.

1. Atr54 - I'm not afraid to tell my spouse about her/his incompetence. (Numeric | Range: 0-4)

Take a look above at the source of the original dataset for more details.

Target (y)

1. Class: (Binary | 1 => Divorced, 0 => Not divorced yet)

School of Computer Science Engineering and Technology

Objective

To gain understanding of logistic regression through implementing the model from scratch

Tasks

1. Download and load the data (csv file contains ';' as delimiter)
2. Add column at position 0 with all values=1 (pandas.DataFrame.insert function). This is for input to the bias
3. Define X matrix (independent features) and y vector (target feature) as numpy arrays
4. Print the shape and datatype of both X and y
5. Split the dataset into 85% for training and rest 15% for testing (sklearn.model_selection.train_test_split function)
6. Follow logistic regression class and fill code where highlighted:
7. Write sigmoid function to predict probabilities
8. Write log likelihood function
9. Write fit function where gradient ascent is implemented
10. Write predict_proba function where we predict probabilities for input data
11. Train the model
12. Write function for calculating accuracy
13. Compute accuracy on train and test data

Further Fun (will not be evaluated)

Play with learning rate and max_iterations

Preprocess data with different feature scaling methods (i.e. scaling, normalization, standardization, etc) and observe accuracies on

both X_train and X_test

Train model on different train-test splits such as 60-40, 50-50, 70-30, 80-20, 90-10, 95-5 etc. and observe accuracies on both X_train

and X_test

Shuffle training samples with different random seed values in the train_test_split function. Check the model error for the testing

data for each setup.

Print other classification metrics such as:

classification report (sklearn.metrics.classification_report),

confusion matrix (sklearn.metrics.confusion_matrix),

precision, recall and f1 scores (sklearn.metrics.precision_recall_fscore_support)

Helpful links

How Logistic Regression works: <https://machinelearningmastery.com/logistic-regression-for-machine-learning>

Feature Scaling: <https://scikit-learn.org/stable/modules/preprocessing.html>

Training testing splitting: [sklearn.model_selection.train_test_split — scikit-learn 1.2.0 documentation](https://scikit-learn.org/stable/modules/model_selection.html)