EE658/758 Machine Learning in Engineering

Assignment #2: Logistic Regression
Due Date: Monday, March 4th, 2024

Logistic regression with non-linear decision boundary

Logistic regression is a linear model for classification. It works well when the decision boundary between classes can be approximated by a straight line (or a plane, in higher dimensions).

However, in cases where the decision boundary is non-linear, we can create new features by applying non-linear transformations to the original features. For example, we can add powers of the original features. This can help in capturing non-linear relationships between the features and the target variable.

This programming assignment will guide you through the process of creating predictive models using logistic regression for a dataset with two classes that are not easily separable by a linear decision boundary. You will explore different approaches, including using a popular machine learning library (scikit-learn) and implementing logistic regression from scratch. Additionally, you will experiment with adding higher-degree features to the dataset to see if the model's performance can be improved.

The Dataset

Assume you have a dataset data.csv with three columns: Feature1, Feature2, and Label. The Label column contains binary values (0 or 1), representing two different classes.

The dataset is not linearly separable, meaning you cannot draw a straight line that perfectly separates the two classes.

Part A: Logistic Regression with Scikit-learn

- Load the dataset and split it into training and testing subsets.
- Use the scikit-learn library to develop a logistic regression model based on the training data. Use only Feature1 and Feature2.
- Plot the data points and the decision boundary of the model.
- Evaluate the model on the testing data and report its accuracy.

Part B: Logistic Regression Without Scikit-learn

- Implement logistic regression from scratch.
- Use the same training and testing data splits as in Part A.
- Develop a logistic regression model based on the training data, using only Feature1 and Feature2.
- Display the cost (loss) as a function of iterations during the training process.

- Plot the data points and the decision boundary of your model.
- Evaluate the model on the testing data and report its accuracy.

Part C: Logistic Regression with Feature Engineering (Scikit-learn)

- Enhance your dataset by adding new features that are at a higher degree of one of the original features (e.g., Feature1^2, Feature1 * Feature2).
- Use the scikit-learn library to develop a logistic regression model based on the enhanced training data.
- Plot the data points and the decision boundary of the model.
- Evaluate the model on the testing data and report its accuracy.

Part D: Logistic Regression with Feature Engineering (Without Scikit-learn)

- Use the enhanced dataset from Part C.
- Implement logistic regression from scratch to develop a model based on the enhanced training data.
- Display the cost (loss) as a function of iterations during the training process.
- Plot the data points and the decision boundary of your model.
- Evaluate the model on the testing data and report its accuracy.