Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining
Fall 2022 (Kerner)

Homework 1: Classification

Part 1

Logistic regression is a probabilistic discriminative model that directly assigns class labels without computing class-conditional probabilities. In this approach, the following ratio of posterior probabilities is sufficient for inferring class labels in a binary classification problem:

$$\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} \qquad \begin{matrix} y=1 \text{ if } > 1 \\ y=0 \text{ if } \leq 1 \end{matrix} \qquad (1)$$

Logistic regression computes the odds using the following equation:

$$\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = e^{z} \qquad (2)$$

where $z = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$ and $\mathbf{x}$ is the input feature vector and $\mathbf{w}$, b are parameters learned during the training procedure. We can manipulate this equation to derive the following logistic function (also called the sigmoid function) used for classification in logistic regression::

$$P(y=1|\mathbf{x}) = \frac{1}{1+e^{-z}} = \sigma(z) \qquad (3)$$

Write the steps to derive Equation 3 from Equation 2.

Submit your answers as a pdf with filename `<lastname>-hw1-lr-derivation.pdf`. Answers should use mathematical notation and should be written neatly by hand (and scanned as pdf) or typed using latex or another math library.

Part 2

In this exercise, you will implement a binary classification model for the Census Income dataset. The task for this dataset is to predict whether an individual's income exceeds $50,000 per year or not based on a set of attributes/features (including age, occupation, education, and other factors). The dataset was constructed from the 1994 US Census database.

You may choose to implement any model that we have covered in class:

- K nearest neighbors
- Naive Bayes
- Support Vector Machine
- Decision Tree
- Random Forest
- Logistic regression
- Neural networks
- Ensemble methods

You may use any libraries that we have used in labs, including Scikit-learn, numpy, pandas, and keras/tensorflow.

Data Preparation

You will split your dataset into a training (60% of the total data) and test (40%) set. You do not need to create a validation set because you will use cross validation for hyperparameter tuning (details in the Model Training section).

You will need to inspect the dataset to determine what data preparation steps need to be applied, which may also vary depending on the model you choose to use. These steps may include:

- Normalization or standardization
- Conversion of categorical data to numerical data or vice versa
- Handling of duplicate and/or missing data
- Feature selection

Model Training

The details of training your model will vary depending on which model you choose to implement. For all models, you will find the optimal hyperparameters of your model using 5-fold cross validation and Grid Search (more details here and in Lecture 11).

Evaluation

Your final model evaluation should be performed on the test set. Report the following metrics on the test set:

- Overall accuracy

- Precision
- Recall
- F1 score

Other considerations

Be sure to set your random seed at the beginning of your code and take any other steps to maximize reproducibility of your model results. Use a random seed = 0 whenever a seed is required; this will maximize consistency across solutions by many students.

Submission

You will add your code to the notebook provided in the assignment instructions which contains starter code for loading the dataset (`cse572-homework1.ipynb`). Rename the notebook to `cse572-homework1-<lastname>.ipynb` and submit the following three deliverables:

1. a link to your Colab notebook (as a comment on the submission)
2. your .ipynb file (`cse572-homework1-<lastname>.ipynb`)
3. a pdf of the executed notebook (`cse572-homework1-<lastname>.pdf`)

Grading

Grading for Part 1 will be based on a correct derivation of Equation 3 from Equation 2. Partial credit will be assigned as appropriate. Be sure to show all steps and provide explanations as needed to ensure you receive full credit.

Grading for Part 2 will be based on your code and the accuracy of your results. You do not have to achieve the best possible accuracy on the test set, but the overall accuracy should exceed random guessing.

Points will be distributed as follows:

- Part 1: 20 points
- Part 2: 30 points for Data Preparation, 30 points for Model Training, and 20 points for Evaluation.