Statistical Pattern Recognition

Homework 1

Linear and Logistic Regression and Multiclass Classification

Assignment Date: 1404/10/3 **Submission Deadline:** 1404/10/16

Objectives and Precautions:

In this homework, you will learn:

- How to better understand and explore a dataset through Exploratory Data Analysis (EDA).
- How to implement linear and logistic regression models and evaluate their performance using appropriate validation techniques.
- How to perform multi-class classification using the logistic regression family of algorithms.

You are required to implement all models **from scratch** and are restricted to using only the following Python libraries in your implementation:

- **Pandas:** For reading and processing datasets.
- **Numpy:** For handling arrays and matrix operations.
- **Matplotlib.pyplot:** For visualizing samples, model outputs, and regression lines.

Additional Notes:

- You have **two weeks** to complete this assignment.
- The assignment instructions and datasets will be provided through your **Quera class.**
- You are allowed to use **only the Python programming language**.
- Save your results and answers in any format of your choice as a report, and submit it together with your code.
- Package your code files and the report into a **single ZIP file** and upload it to Quera. The ZIP file should be named using the format **"YourNames.zip"**.
- If you have any questions, feel free to ask them in the **Telegram channel**
- Ensure that all submitted work is **original and written entirely by you**; copying from others is strictly prohibited.
- You should be prepared to **clearly explain every part of your code** to the Teaching Assistants, especially if external tools (such as ChatGPT) were used for guidance.

Late Submission Policy

- Submissions delayed by **one day** will receive **70%** of the total score.
- Submissions delayed by **two days** will receive **50%** of the total score.
- Submissions submitted **more than two days late will not be graded**.

Good luck!

1. Linear Regression

Linear regression models the relationship between a dependent variable y and one or more independent variables x using a linear equation. In this section, you will implement **Simple Linear Regression** from scratch and study different optimization strategies.

1.1. Understanding the Dataset: auto_mpg.csv dataset

In this part, we use the **Auto MPG** dataset, which contains technical specifications of automobiles manufactured in the 1970s and early 1980s. The objective is to predict a car's fuel efficiency, measured in miles per gallon (mpg), based on vehicle and engine characteristics.

Dataset Features:

- **mpg:** Fuel efficiency of the car (target variable, continuous)
- **weight:** Vehicle weight
- **horsepower:** Engine power
- **displacement:** Engine displacement

1.2. Tasks to Understand the Dataset: auto_mpg.csv

1. Load the Auto MPG dataset using pandas.read_csv().
2. Take a look at some samples of your dataset using (e.g., data.sample(8))
3. How many samples are there in the dataset? (you may use data.shape)
4. Compute and report the mean and standard deviation of the target variable **mpg**.
5. Compute the correlation between **weight** and **mpg**, and explain the concept of correlation as well as the meaning of its sign and magnitude in this context.
6. Plot a scatter plot of **weight** versus **mpg**, and repeat this visualization for the other features. Briefly discuss the observed relationship between vehicle characteristics and fuel efficiency.

1.3. Creating the Model: Simple Linear Regression

Simple linear regression involves one dependent variable and one independent variable. Therefore:

1. Retain only the weight feature as the input variable and mpg as the target variable, and drop all other features from the dataset.
2. Split the dataset into training and testing sets using an 80–20 split with train_test_split and random_state = 42. Report the number of samples in each set.
3. Normalize the **weight** values in the training set to have mean 0 and standard deviation 1, and use the same training-set parameters to normalize the test set.

You must implement linear regression **from scratch** and train the model using the following methods.

A) Batch Gradient Descent (BGD)

1. Implement Batch Gradient Descent for linear regression and train the model on the training set.
2. Plot the cost function (MSE) versus the number of iterations (record the cost every 100 iterations).
3. Experiment with different learning rates and numbers of iterations, and report your observations.
4. Report the learned model parameters θ_0 and θ_1.
5. Plot the regression line together with the training and testing samples.

B) Stochastic Gradient Descent (SGD)

1. Implement Stochastic Gradient Descent, where model parameters are updated after each training sample.
2. Train the model using the same training set and feature normalization as in Batch Gradient Descent.
3. Plot the cost function over training epochs.
4. Experiment with different learning rates and discuss their effect on the convergence behavior.
5. Report the learned parameters θ_0 and θ_1.

C) Comparison of Batch GD and SGD

Compare Batch Gradient Descent and Stochastic Gradient Descent in terms of:

1. Speed of convergence
2. Stability and smoothness of the cost function
3. Final cost value (or MSE)
4. Sensitivity to learning rate

Explain the observed differences based on the update rules of the two optimization methods.

D) Closed-form Solution

Implement linear regression using the closed-form solution (normal equation).

1. Report the obtained parameters θ_0, θ_1.
2. Plot the regression line along with the training and testing samples.
3. Compare the closed-form solution with the results obtained using Batch GD and SGD.

---

2. Logistic Regression

In contrast to linear regression that aims to predict a continuous output, logistic regression is used for binary classification problems. It models the probability that a sample belongs to one of two possible classes.

2.1. Understanding the Data: Bank Marketing

In the Bank Marketing Dataset each sample represents a bank client described by numerical and categorical features related to demographic information and marketing campaign attributes. The target variable y indicates whether the client subscribed to a term deposit (yes) or not (no). The goal of this problem is to build a binary classification model to predict client subscription.

2.2. Tasks to Understand the Dataset: bank-full.csv

Follow the steps below to explore and prepare the dataset for building a logistic regression model.

1. Load the dataset. (You can use pandas.read_csv() to read the dataset)
2. How many samples are there in total? How many features does the dataset contain?
3. Report the number of numerical and categorical features using data.info().
4. Convert categorical features into numerical form using appropriate encoding techniques:
   - Apply ordinal encoding only to categorical features that have a true and meaningful order.
   - Use one-hot encoding for nominal categorical features where no inherent ordering exists.

   Explain why encoding categorical variables is necessary for logistic regression, and discuss how inappropriate encoding may introduce artificial relationships between categories.
5. Is the dataset balanced or imbalanced?

6. After encoding the categorical features, compute the correlation between each feature and the target variable as a heuristic measure of association. Select the seven features with the highest absolute correlation values and retain only these seven features along with the target column, and drop all other features from the dataset. Then check for duplicate samples in the training set and remove them if necessary.
7. Since the selected features may have different numerical scales, standardize only the numerical features to have zero mean and unit variance.
8. Plot the histogram of each selected numerical feature after standardization. Comment on the shape and distribution of the features.
9. Split the prepared dataset into training (80%) and testing (20%) sets using stratified random sampling (with random_state=42) to preserve class proportions. Then separate both sets into features (X) and labels (y). Report the shape of X_{train}, $X_{test}, y_{train}, and\ y_{test}$.

2.3. Steps to Build a Logistic Regression Model:

1. Implement logistic regression from scratch, without using any machine learning libraries, and train it on the training set.
2. Compute and plot cost function at each iteration to analyze the convergence behavior of the training process.
3. Train the model using multiple learning rates and different numbers of epochs, and compare their effects on convergence speed and final performance.
4. Report all learned model parameters (θ values), including the bias term, and discuss the role of each parameter in the logistic regression model.
5. Use the trained model to classify samples in both the training and test sets. Compute and report the accuracy, precision, recall, and F1-score. Define and explain each metric.
6. Compute the confusion matrix for the test set and explain its importance in evaluating classification performance, particularly in terms of correctly and incorrectly classified samples.

---

3. Multi-class Classification using Logistic Regression

Discriminative multi-class classification aims to learn decision boundaries that effectively separate multiple classes within a dataset. Three common methods for discriminative multi-class classification are one-vs-one, one-vs-all and softmax regression.

- One-vs-One (OvO) trains several binary logistic classifiers, each specialized in distinguishing between a specific pair of classes, and predicts the class based on their combined outcomes.
- One-vs-All (OvA) builds one binary classifier per class against all remaining classes and assigns the label with the highest confidence score.
- Softmax regression extends logistic regression to the multi-class setting by modeling a normalized probability distribution over all classes and selecting the class with the maximum probability.

3.1. Understanding the Data: Wine Dataset

The Wine dataset contains chemical analysis results of wines produced from three different grape cultivars in the same region, forming a multiclass classification problem. Each sample is represented by chemical and physical features such as alcohol content, acids, phenolic compounds, and color properties that describe the wine's composition and sensory characteristics. These features were obtained through standard laboratory analyses, and the goal is to classify wines into their corresponding classes based on the measured attributes

3.2. Tasks to Understand the Dataset: wine_dataset.csv

1. Report the mean and standard deviation of each feature for descriptive purposes only, without using these statistics in model training or preprocessing steps.
2. If categorical features exist, convert them to numerical representations using label encoding.
3. Use a stratified split to preserve class proportions, with 75% of the data for training and 25% for testing.
4. Detect and handle missing values and duplicated samples using statistics computed only from the training set, and apply the same cleaning rules to the test set without recomputing any statistics.
5. Detect outliers using Z-score (threshold = 2.75), remove them, and report how many were removed
6. Normalize features using min-max normalization based on parameters computed from the training set, and apply the same transformation to the test.
7. Explain the difference between normalization and standardization.

3.3. Creating Model: OvO, OvA, Softmax Regression

1. Implement OvA and OvO algorithms to classify your multi-class dataset. Use logistic regression as the main classifier.
2. Train the OvA and OvO models and report both training and test accuracy, ensuring that the evaluation is performed on a held-out test set that was not used in any preprocessing or model training steps.
3. For each binary logistic regression model trained within the OvA and OvO frameworks, plot the cross-entropy loss (cost function) over training iterations and report the iteration at which convergence is achieved, based on a predefined convergence criterion.
4. Implement softmax logistic regression as a direct multi-class classification model, using the multiclass cross-entropy loss function. Report the training and test accuracy, and plot the evolution of the loss function during training.
5. Compare the performance of OvA, OvO, and softmax regression in terms of test accuracy, convergence behavior, and computational complexity, and identify which approach yields the best overall performance on the dataset.
6. Evaluate the impact of outlier removal on classification performance by training and testing all three models both with and without the detected outliers removed from the training set, and analyze whether omitting outliers leads to improved generalization.

---

4. Bonus: Regularized Linear Regression with SGD

In this bonus section, you will extend your linear regression model by applying **Ridge Regression (L2 regularization)** and training it using **Stochastic Gradient Descent (SGD)** on the **auto_mpg dataset**. The goal is to study the effect of **multicollinearity, regularization, and optimization stability** in linear models.

1. **Multicollinearity Analysis (Before Regularization):** Compute the correlation matrix of all input features. Calculate the **Variance Inflation Factor (VIF)** for each feature and identify highly correlated predictors.
2. **Ridge Regression with SGD:** Add an L2 regularization term to the linear regression cost function (do not regularize the bias term). Train the model using **SGD** for multiple values of the regularization parameter λ (e.g., 0, 0.01, 0.1, 1). Use reasonable hyperparameters of your choice (e.g., learning rate, number of epochs, batch size) and report them.
3. **Stability and Performance Comparison:** For each value of the regularization parameter λ, compare the learned model coefficients with those obtained from the unregularized model. Plot the training loss (MSE) versus training epochs to analyze the convergence behavior and stability of SGD. Additionally, report and compare the final training and testing Mean Squared Error (MSE) for all λ values.
4. **Analysis and Discussion:** Explain how Ridge Regression mitigates the negative effects of multicollinearity. Discuss the observed **bias–variance tradeoff** as λ changes.