In the Name of God

دانشگاه شیراز
Shiraz University

Statistical Pattern Recognition

Home Work I

Linear and Logistic Regression and Multiclass Classification

Assignment Date: 27 Aban

Submission Deadline: 11 Azar

1. Overview:

Goals of this assignment:

1. **Linear Regression**:
    - Model relationships between features and continuous outcomes.
    - Optimize and evaluate regression performance.
2. **Logistic Regression**:
    - Address binary classification using predictive models.
    - Analyze model effectiveness with standard metrics.
3. **Multi-class Classification**:
    - Develop strategies for multi-class problems like ova, ovo and softmax.
    - Compare and refine classification techniques.
4. **Model Analysis**:
    - Optimize models and assess performance impact.
    - Draw insights from various modeling approaches.

Precautions:

You are required to write the models from scratch and you are restricted to using only the following three Python libraries in your implementation:

- **Pandas**: For reading and processing datasets.
- **Numpy**: For handling arrays and matrices.
- **Matplotlib.pyplot**: For visualizing your samples and regression lines.

Additionally:

- You have **two weeks** to complete this assignment.
- The assignment details and datasets will be provided in your Quera class.
- You are allowed to use only the Python programming language.
- Save your results and answers to the questions in any format of your choice as your report and upload it with the code.
- Package your code files and report into a single zip file and upload it to Quera. The naming format for the zip file should be `your names.zip` (e.g., `Kylian_Mbappe_Erling_Haaland.zip`).

- **Late submission policy**: Submissions delayed by one day will receive 70% of the total score; those delayed by two days will receive 50%. No submissions will be graded after two days.
- If you have any questions, feel free to ask them in the Telegram channel.
- Ensure that your work is original and written entirely by you; avoid copying from others.
- Be prepared to explain every part of your code to the Teaching Assistants, especially if you use tools like ChatGPT for guidance.

2. Linear Regression

Linear regression models the relationship between a dependent variable (y) and one or more independent variables (x) as a linear equation. The change in the dependent variable is assumed to be directly proportional to the change in the independent variables.

2.1 Understanding the Dataset: `Advertising.csv:`

The Advertising dataset examines the relationship between advertising expenditures and product sales across television, radio, and newspaper media. The dataset includes:

- Sales amounts (in thousands of units)
- Budget allocated to TV, radio, and newspaper advertisements (in thousands of dollars)

Tasks to Understand the Dataset:

1. Load `Advertising.csv` using `pandas.read_csv()`.
2. Display five random samples of the dataset using `dataset.sample(5)`.
3. Print the number of samples using `dataset.shape[0]`.
4. Compute and report the mean value of the target variable (sales) and its standard deviation.
5. Identify which feature has the strongest linear relationship with the target variable using correlation analysis. Explain correlation and covariance.
6. Plot scatter plots of TV values vs. sales and repeat for other features. Discuss if sales increase with TV values.

Summary: Identify the variables with the strongest linear correlation with sales and examine scatter plots for insights.

2.2 Creating Our Model: Simple Linear Regression

Simple linear regression models the relationship between one dependent and one independent variable.

Steps to Create Your Model:

1. Retain only the `TV` and `sales` variables in the dataset; drop the rest.
2. Split the dataset into training (80%) and testing sets using `train_test_split` from `sklearn.model_selection` with `random_state=42`.
3. Report the number of samples in the training and testing sets.
4. Normalize `TV` values in the training set to have a mean of 0 and a standard deviation of 1. Use training set parameters to normalize the test set.

Linear Regression Implementation:

1. Implement linear regression using the gradient descent algorithm (batch or stochastic) and train on the training set.
2. Plot the cost function over iterations, sampling every 100 iterations.
3. Experiment with different learning rates and iterations, and report your findings.
4. Report the best parameters for your regression line (e.g., `theta_0`, `theta_1`).
5. Plot the regression line with training and testing samples.
6. Implement linear regression using the closed-form solution and repeat steps 4 and5.

3. Logistic Regression

Logistic regression models the probability of a binary outcome based on one or more predictor variables. The dependent variable must be binary, and observations should be independent.

3.1 Understanding the Dataset: `weatherAUS.csv`

The weather dataset contains daily weather observations from Australia, including 23 numerical and categorical attributes and a binary class label (`RainTomorrow`).

Tasks to Understand and Prepare the Dataset:

1. Load the dataset.
2. Display samples and report the number of numerical, categorical, and binary features using `data.info()`.
3. Identify the seven features with the highest correlation to the target.
4. Find features with missing values and check for duplicates.
5. Remove samples with missing values and report how many were removed.
6. Convert categorical features to numerical using label encoding. Explain why this step is necessary.
7. Retain the seven most correlated features and the target variable. Report the dataset shape.

3.2 Creating Our Model: Logistic Regression

Steps to Build the Model:

1. Split the dataset into training (80%) and testing sets using `train_test_split` with `random_state=42`.
2. Implement logistic regression from scratch and train on the training set.
3. Plot the cost function over iterations.
4. Experiment with different learning rates and iterations, and report your findings.
5. Report all parameters (`theta`) for the model.
6. Classify samples in both sets and calculate accuracy, precision, recall, and F1-score. Define and explain each metric.
7. Compute the confusion matrix and explain its importance in classification.

---

4. Multi-class Classification using Logistic Regression

Discriminative Multi-class Classification techniques include one-vs-one (OvO), one-vs-all (OvA), and softmax regression.

4.1 Classification Using OvA

OvA involves training a binary classifier for each class against the rest and selecting the class with the highest confidence.

Dataset: `load_risk_dataset.csv` The dataset includes features like age, gender, education, income, and risk levels (low, medium, high).

Tasks:

1. Load the dataset.
2. Report the number of samples and the mean and standard deviation of each feature.
3. Convert categorical features to numerical using label encoding.
4. Normalize features using min-max normalization. Explain the difference between normalization and standardization.
5. Check for missing values and duplicates; remove affected samples.
6. Detect outliers using Z-score (threshold = 2.6), remove them, and report how many were removed. Experiment with different thresholds and explain your findings.

4.3.1 Creating Our Model: OvA Classification

1. Split the dataset into training (80%) and testing sets.
2. Report the number of samples in each set.
3. Implement the OvA model using logistic regression code from earlier sections.
4. Plot the cost function for each logistic regression and report convergence iterations.

4.3.2 Creating Our Model: Softmax

1. Implement Softmax regression and report training and testing accuracy. Plot the cost function.
2. Compare OvO, OvA, and Softmax methods, discussing performance differences.
3. Re-run classification without removing outliers and discuss the impact on accuracy.

5. Bonus: OvO Classifier

1. Implement the OvO classifier and train the model.
2. Report all requested metrics and comparisons.
3. Compare OvA and OvO methods. Discuss differences and your interpretations.